CHARLES UNIVERSITY

FACULTY OF SOCIAL SCIENCES

Institute of Economic Studies

Can Model Combination Improve Volatility Forecasting?

Master's thesis

Author: Bc. Sabyrzhan Tyuleubekov

Study program: Economics and Finance

Supervisor: doc. PhDr. Jozef Baruník, Ph.D.

Year of defense: 2019

Declaration of Authorship

The author hereby declares that he or she compiled this thesis independently, using only the listed resources and literature, and the thesis has not been used to obtain any other academic title.

The author grants to Charles University permission to reproduce and to distribute copies of this thesis in whole or in part and agrees with the thesis being used for study and scientific purposes.

Prague, July 29, 2019

Sabyrzhan Tyuleubekov

Abstract

Nowadays, there is a wide range of forecasting methods and forecasters encounter several challenges during selection of an optimal method for volatility forecasting. In order to make use of wide selection of forecasts, this thesis tests multiple forecast combination methods. Notwithstanding, there exists a plethora of forecast combination literature, combination of traditional methods with machine learning methods is relatively rare. We implement the following combination techniques: (1) simple mean forecast combination, (2) OLS combination, (3) ARIMA on OLS combined fit, (4) NNAR on OLS combined fit and (5) KNN regression on OLS combined fit. To our best knowledge, the latter two combination techniques are not yet researched in academic literature. Additionally, this thesis should help a forecaster with three choice complication causes: (1) choice of volatility proxy, (2) choice of forecast accuracy measure and (3) choice of training sample length. We found that squared and absolute return volatility proxies are much less efficient than Parkinson and Garman-Klass volatility proxies. Likewise, we show that forecast accuracy measure (RMSE, MAE or MAPE) influences optimal forecasts ranking. Finally, we found that though forecast quality does not depend on training sample length, we see that forecast combination methods outperform standalone methods on a longer training sample. Finally, we found that KNN regression on OLS combined fit on medium training sample outperforms other methods for Garman-Klass volatility estimate.

JEL Classification C53, C58, E37, G17, G32
Keywords model combination, time series, volatility, forecast, machine learning

Title Can Model Combination Improve Volatility Forecasting?

Author's e-mail styuleubekov@gmail.com
Supervisor's e-mail barunik@fsv.cuni.cz

Abstrakt

V současné době existuje řada metod predikce a prognostici čelí mnoha výzvám při výběru optimální metody pro predikci volatility. Tato diplomová práce testuje několik metod kombinací predikce, aby bylo možné využít široké škály prognóz. Bez ohledu na to, že existuje spousta literatury o kombinaci prognóz, kombinace tradičních metod s metodami machine learning je relativně vzácná. V této práci implementujeme následující kombinované metody: (1) simple mean forecast combination, (2) OLS combination, (3) ARIMA on OLS combined fit, (4) NNAR on OLS combined fit a (5) KNN regression on OLS combined fit. Na základě námi dostupných informací nejsou poslední dvě kombinované metody doposud zkoumány v akademické literatuře. Tato práce by navíc měla pomoci prognostici se třemi možnými komplikacemi: (1) výběr volatility proxy, (2) výběr měřítka přesnosti predikce a (3) výběr délky zkušebního vzurku. Zjistili jsme, že squared a absolute return proxy jsou mnohem méně účinné než Parkinson a Garman-Klass volatility proxy. Dále ukazujeme, že metriky přesnosti prognózy (RMSE, MAE nebo MAPE) ovlivňují pořadí optimálních prognóz. Dalším zjištěním je, že přestože kvalita predikce nezáleží na délce zkušebního vzorku, je vidět, že metody kombinace predikcí překonávají samostatné metody na delších zkušebních vzorcích. Na závěr jsme zjistili, že Garman-Klass volatiltiy proxy, KNN regression on OLS combined fit na střední délce zkušebního vzorku překonává jiné metody pro estimaci Garman-Klass volatility.

Klasifikace JEL	C53, C58, E37, G17, G32
Klicova slova	model combination, time series, volatility, forecast, machine learning

Nazev prace	Může modelová kombinace řídit prognózu volatility?

E-mail autora	styuleubekov@gmail.com
E-mail vedouciho prace	barunik@fsv.cuni.cz

Acknowledgments

The author is grateful to doc. PhDr. Jozef Baruník, Ph.D.

Typeset in LATEXusing the IES Thesis Template.

Bibliographic Record

Tyuleubekov, Sabyrzhan: *Can Model Combination Improve Volatility Forecasting?*. Master's thesis. Charles University, Faculty of Social Sciences, Institute of Economic Studies, Prague. 2019, pages 110. Advisor: doc. PhDr. Jozef Baruník, Ph.D.

Contents

List of Tables

List of Figures

Acronyms

AIC(c)	Akaike Information Criterion (with correction)
(A)NN	(Artificial) Neural Netoworks
AR(F)IMA	Auto Regressive (Fractionally) Integrated Moving Average
ERLS	Equality Restricted Least Squares
EWMA	Exponentially Weighted Moving Average
(G)ARCH	(Generalized) Auto Regressive Conditional Heteroscedacticity
kNN	k Nearest Neighbours
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multi-layer Perception
NARX	Nonlinear Autoregressive with eXogenous approach
NRLS	Non-negativity Restricted Least Squares
(P)ACF	(Partial) Auto-Correlation Function
RMSE	Root Mean Square Error
RW	Random Walk
(S)BIC	(Schwarz's) Bayesian information criterion
SMA	Simple Moving Average
SVR	Suppoty Vector Regression

Master's Thesis Proposal

Author	Bc. Sabyrzhan Tyuleubekov
Supervisor	doc. PhDr. Jozef Baruník, Ph.D.
Proposed topic	Can Model Combination Improve Volatility Forecasting?

Motivation Volatility forecast is a very popular topic among economists since it is widely used by practitioners: risk modellers, stock traders, investors, etc. for estimation of potential risks (e.g. VaR), pricing financial derivatives (e.g. option prices) construction of optimal portfolios (e.g. estimation of Sharpe ratio), etc. For these purposes, some fundamental time series analysis models were designed, such as AR(I)MA (popularized by G. Box and G. Jenkins in 1970), ARCH (by R. Engle in 1982), GARCH (by T. Bollerslev in 1986) etc. These models and their extensions are widely studied and used in academia. There is a lot of research of time series on volatility forecasting, yet no consensus has been reached what model is the best, for instance, comparative analyses were done by Einarsen (2014) and Poon and Granger (2003). On the other hand, we have literature supporting model combination: Aiolfi et al (2010), Degiannakis (2017), etc. Model combination is useful in situations when combined forecast from several models result in a better forecast than forecasts from each model on their own, so combination can reduce errors. Besides conventional models such as mentioned above and their extensions, we also would like to implement machine learning and neural network techniques. There are some similar researches done: e.g. De Stefani et al (2017), Andre et al (2017), however much less than using conventional time series analysis methods. Therefore, we want to do a research that would combine forecast from conventional models and machine learning and neural network methods resulting in a more accurate volatility forecast than each model on their own.

Hypotheses

Hypothesis #1: Forecast from a combined model is more accurate than forecasts from conventional models or their extensions.

Hypothesis #2: Accuracy of volatility forecast for less risky assets is higher than for more risky ones.

Hypothesis #3: Volatility forecast for less risky assets and more risky ones need different model combination.

Methodology Firstly, we are planning to introduce you with main popular methods for volatility forecast and then run various forecasts using (F)AR(I)MA, GARCH and GARCH extensions (e.g. such as FIGARCH, EGARCH, TGARCH, etc.) additionally we are planning to use SMA and EWMA, similar work was already carried out by Einarsen (2014) where he did a comparative study of volatility forecasting models, likewise a broad review of volatility forecasts was done by Poon and Granger (2003). In addition, we will do forecasts with different horizons and will try to find which horizon is optimal based on accuracy, Brownlees et al (2011) were assessing forecasts done by ARCH class of models with different horizons. All our forecasts will be out-of-sample and will be tested on 3 types of datasets: (1) risky assets, (2) safe assets and (3) mixed. Decision on whether asset should be classified as risky or safe will be based on historical volatility, likewise we will try to apply unsupervised learning to classify our data (clustering methods such as h-clust and k-means). Also, we will try to run forecasts using machine learning techniques and neural networks like De Stefani et al (2017), Andre et al (2017), Basavaraj (2015) and Ladokhin (2009) where they implemented artificial neural networks (ANN), k-Nearest Neighbours (kNN) and support vector regression (SVR) etc.

After we run forecasts using different models we will evaluate their forecasts using different horizons. Metrics for forecast assessment will be RMSE, MAE, MAPE, ME and MPE.

Afterwards, we will combine forecast models that showed the best accuracy and see whether a combined forecast has advantages towards single model forecasts, similar research was already done by Aiolfi et al (2010), Timmermann (2018), Degiannakis (2018), Diebold (1988) and Bates and Granger (1969). Same forecast assessment as with single models will be done.

Finally, we will compare forecasts and will try to draw some conclusions and test our hypotheses.

All three hypotheses written above are straightforward to test.

Expected Contribution We expect to find the best model for risky and safe assets and also define optimal horizons when forecast predictions are most accurate. Likewise, we expect to find such forecast combination that would outperform forecasts from other popular models. We hope, our results could be directly used in practice by implementing our methods in volatility forecast. There are few researches that combine both conventional methods and machine learning/neural networks methods in volatility forecasts.

Outline

1. Introduction – where we explain motivation of the work and why it is interesting nowadays, since with increased computational power of computers we can now implement different models, likewise with increasing popularity of machine learning and neural networks we can bring new results.
2. Literature review – where we will summarise already published works on model combination and single forecast models.
3. Data – where we will describe our data and how we classify it, likewise, we will implement unsupervised learning techniques in order to classify our data in risky and safe assets.
4. Methods – where we will describe all the models we are going to use in the paper and how we will combine forecasts.
5. Results – we will compare forecasts for all the models for all classes of assets and for various time horizons and will see which are best.
6. Conclusion – we will summarise our work and will make some notes for future research.

Core bibliography

Einarsen, R. H. (2014) A Comparative Study of Volatility Forecasting Models.

Brownlees, C., Engle, R., Kelly, B. (2011) A practical guide to volatility forecasting through calm and storm. The Journal of Risk (3â€“22).

Aiolfi, M., Capistran, C., Timmermann, A. (2010) Forecast Combinations.

Mitri, M., Gauge, V. (????) Forecasting Equity Realized Volatility using Machine Learning Methods.

Timmermann, A. (2018) Forecasting Methods in Finance.

Poon, S., Granger C. W. J. (2003) Forecasting Volatility in Financial Markets: A Review. Journal of Economic Literature pp. 478-539.

Ladokhin, S. (2009) Forecasting Volatility in the Stock Market.

De Stefani, J., Caelen, O., Hattab, D., Bontempi, G. (2017) Machine Learning for Multi-step Ahead Forecasting of Volatility Proxies.

Degiannakis, S. (2017) Multiple days ahead realized volatility forecasting: Single, combined and average forecasts. Global Finance Journal 36 (2018) 41-61.

Diebold, F. X. (1988) Serial Correlation and the Combination of Forecasts. Journal of Business & Economic Statistics, January 1988, Vol.6, No. 1.

Bates, J.M. and C.W.J. Granger (1969) The Combination of Forecasts. Operations Research Quarterly 20, 451-468.

Reider, R. (2009) Volatility Forecasting I: GARCH Models.

Reider, R. (2009) Volatility Forecasting II: Stochastic Volatility Models and Empirical Evidence.

Hemanth Kumar, P., Basavaraj Patil, S. (2015) Volatility Forecasting using Machine Learning and Time Series Techniques. International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 9, September 2015.

Andre, J., Wechselberger, A., Zhao, S. (2017) Volatility Forecasting using SVM.

Author

Supervisor

Chapter 1

Introduction

This thesis should provide you insights about volatility forecasting, forecasts combinations and volatility proxies.

Volatility is an important topic in financial markets since volatility plays a crucial role in various decision-making processes related to financial investments and policies because volatility is a fundamental measure of risk. Therefore, volatility is a crucial part of derivatives pricing, Value-at-Risk estimation, optimal portfolio construction and etc. High volatility is associated with higher risk, so volatility also indirectly influences investment policies of various institutions since institutions that participate in financial markets (e.g. pension funds, asset management companies) often have strict rules regarding maximum tolerable risk for an investment. For example, pension funds or government institutions usually can invest only in investment-grade bonds and rating (grade) of a bond can be affected by a volatility of a stock price of an institution issuing the bond. Likewise, banks that have trading activity must allocate some budget for covering market risk and this buffer influences key performance indicators of a bank (e.g. return on equity). Therefore, need for accurate volatility estimation and forecast is also regulatory driven in the world of financial markets and due to this academia also has been researching this topic thoroughly.

All institutions that are affected by the need of precise volatility estimation and forecasts endeavor to find the most accurate forecast. However, nowadays, we have a lot of forecasting methods to select from and often it is difficult for a forecaster to determine an optimal one. There are multiple reasons that can complicate the choice of the best forecasting method and in this thesis we concentrate on tackling three complication causes[1]: (1) choice of volatility proxy, (2) choice of forecast accuracy measure and (3) choice of training sample length.

Volatility proxy choice is dependent on what type of data a forecaster has. In the era of high frequency data, many forecasters would prefer to build their forecasts on it. However, often high frequency data is not easily (or freely) available, also, models trained on high frequency data require more computational power than models trained on data with lower sampling frequency (holding time bounds of training sample constant). The most easily available data type is daily data, which is freely accessible at most online platforms such as Yahoo finance or Google finance. With daily data it is not straightforward on how to approximate volatility in order to have also volatility estimates at daily frequency because most online platforms provide only four price points (opening, closing, highest and lowest price points) for free. Therefore, volatility proxy is needed because sample standard deviation is extremely imprecise given sample consists of maximum four observations per period. Many academicians utilize squared or absolute return volatility proxies in their researches, whereas in our opinion, both these proxies are unacceptably noisy and forecast quality of these two proxies are low. Therefore, we analyze another two proxies: Parkinson and Garman-Klass volatility proxies, which are price-range based proxies, and compare them with volatility proxies of squared and absolute return. Parkinson (1980) utilized information about highest and lowest price points for a daily volatility proxy and Garman & Klass (1980) extended Parkinson proxy by incorporating also information about opening and closing prices.

Second complication for a forecast selection is how to measure forecast quality. Forecast quality can be measured by various methods that can lead to inconclusive conclusions. For example, Root Mean Square Error (RMSE) may conclude that forecast from model A is the most accurate and at the same time, Mean Absolute Error (MAE) may indicate that model B produces the most accurate forecast. Thus, a forecaster would have multiple metrics that suggest different choices.

[1] Originally, we wanted test three hypotheses outlined in Master's Thesis Proposal, however, during the course of the research we decided to drop hypotheses #2 and #3 and substitute them with the following ones:

> Hypothesis #2 — Different training sub-samples require different models for same assets.
>
> Hypothesis #3 — Training sub-sample increase deteriorates forecast accuracy.

Third complication — choice of training sample length — is dependent on several factors. One of them being computational efficiency and second one being data relevance. Some models, especially neural networks, require a lot of computational power in order to produce a forecast if trained on a large sample. Originally, we wanted to have three training sub-samples with the longest having more than 2000 observations, but when we started to train models we realized it would take us too long in order to produce needed amount of forecasts and we kept only two training sub-samples (750+ and 250+ observations). A training sample needs to be relevant enough and cover periods of high and low volatility regimes for a stock in order not to provide over-optimistic (in case a model was trained on data of low volatility regime only) or over-pessimistic (in case a model was trained on data of high volatility regime only).

In order to alleviate a forecast choice decision, we decided to try forecast combination methods. Though since forecast combination was popularized by Bates & Granger (1969), there were many researches done on this topic, we decide to combine traditional univariate time-series models with machine learning models that is a relatively rare in academia.

In this thesis, we present comparison of traditional time-series models, machine learning time-series models and models from forecast combinations for four volatility proxies: square return, absolute return, Parkinson and Garman-Klass. Additionally, we compare results of models trained on two training sub-sample sizes: medium and short. Finally, we also compare results for two forecast horizons: 5 and 10 periods ahead. In order to obtain some statistically sound results, we carry out the above on a sample of 20 stocks.

The remainder of the thesis is structured in the following way: Chapter 2 discusses volatility proxy issues, traditional volatility forecasting methods, unconventional time-series forecasting methods and forecasts combination. Chapter 3 describes what data we use and how it is segmented as well as what volatility proxies we use. Chapter 4 describes methods and models we use for forecasting and this chapter is segmented in same way as Chapter 2 with additional section where forecasts assessment and comparison methods are described. Chapter 5 is divided in two subsections. Subsection 5.1.1 provides an illustrative example of what results we obtain for each stock, the example is done for one stock, one training sub-sample, one volatility proxy and two forecast horizons. Since we have a lot of results, we aggregate them and present aggregated results in Subsection 5.1.2. Chapter 6 summarizes aggregated results and main outcomes. Chapter 7 concludes the thesis.

Chapter 2

Literature Review

Since we are going to combine different models of volatility forecasting, we decided to divide literature review in three subsections: traditional volatility forecasting, unconventional volatility forecasting and forecast combination. However, in these subsections we can refer to literature that describes techniques for other time series, i.e. not only volatility. Additionally, we devote part of literature on volatility forecasting to discussion of volatility proxies.

2.1 Literature on volatility forecasting

Volatility forecasting topic has been in focus of academia for a long time and there are many papers on this matter because volatility forecasting is of crucial interest for a wide range of both academicians and practitioners (policy makers, financial derivative analysts, market risk managers, etc.). Volatility in finance can be defined in multiple ways since it is an implicit measure, i.e. you do not observe it in a same way as, for example, we observe closing stock prices. Various volatility proxies have its advantages and drawbacks and in literature there is no clear conclusion on what volatility proxy is the most efficient one.

The most common definition of volatility in academia is standard deviation, that is a square root of a sample variance. Such estimate can be obtained if we have a reasonable amount of observations per unit of time and more we have more efficient and accurate estimate we obtain. This aspect will be discussed in more details later since we do not have many observations per unit of time (which is one day in our case). Figlewski (1997) points out that sample mean[1], especially in small samples, is a very inaccurate estimate of the

[1] That is used for sample variance calculation

true mean and may lead to a biased volatility estimate. Likewise, if you have only daily data and you need to estimate daily volatility, standard deviation as a volatility estimate will be extremely inaccurate because sample size will be up to 4 (usually, opening, closing, highest and lowest price points are available in daily stock data). In order to tackle this issue various volatility proxies for daily data were introduced.

High-low estimate of volatility by Parkinson (1980) was developed and Garman & Klass (1980) extended Parkinson's estimate. Parkinson volatility estimate incorporated information of highest and lowest price points during a trading day for daily volatility estimation. Advantage of Parkinson estimate is that most of easily-accessible data sources (e.g. Yahoo Finance, Google finance, etc.) provide daily highest, lowest, opening and closing prices. Moreover, calculation of this proxy is simple and does not require complex data manipulations. Parkinson estimate is based on an assumption that return is conditionally normally distributed and consequently this is its main drawback since we know that financial data is not normally distributed and generally have fat tails (i.e. have leptokurtic distribution). In order to use this estimate it is advised to apply some trimming techniques to entire data set before running the estimation. Likewise, Parkinson estimate does not account for cases when market is closed and trading is not done, therefore volatility is underestimated due to pauses in trading for more than one day. Garman & Klass (1980) incorporated information about opening and closing prices to Parkinson estimate. Nevertheless, Garman-Klass volatility proxy underestimates volatility because it ignores overnight jumps.

In academia, the most commonly used proxy for daily volatility (before high frequency data became widely available) was squared return. However, squared return is an extremely noisy estimate of volatility. Additionally, this proxy does not account for asymmetrical distribution in sufficient way. Lopez (2001) finds that squared return is an inaccurate estimate of variance due to its asymmetric distribution. Another option for approximating volatility is the use of absolute returns. Davidian & Carroll (1987) show that such volatility proxy is more robust against asymmetry and non-normality. McKenzie (1999) finds that volatility forecasts with absolute returns as proxy for volatility provide more precise forecasts than models with squared returns as volatility proxy.

Andersen *et al.* (1999) find that generally higher data sampling frequency (holding forecast horizon constant) increases forecast quality[2]. Nonetheless,

[2] Andersen *et al.* (1999) used intraday 5-min returns for GARCH volatility estimation.

with forecast horizon getting longer lower data sampling frequency works better. This is because with high frequency data and long horizon (e.g. more than six months (Alford & Boatsman (1995)) volatility mean reversion is difficult to adjust.

Likewise, due to complex structure of volatility, its forecast quality will be different depending on current volatility regime and overall volatility level (Diebold *et al.* (1998)). Additionally, Diebold *et al.* (1998), claim that optimal volatility forecasting horizons depend on the underlying asset class and industry.

All forecasting methods we are using in this paper are based on historical data only, i.e. we are using only volatility estimates and no external variables. However, it should be mentioned that there exists a method of implicit volatility forecasting from option prices.

Poon & Granger (2003) carried out a comprehensive comparative study of volatility forecasting models. Their study covered 93 papers on volatility forecasting and compared models performance and what was the volatility proxy used. Poon & Granger (2003) segment volatility forecasting methods in 4 groups:

- HISVOL – historical volatility models, that include: random walk, historical averages, moving averages, AR(F)IMA models,
- GARCH – any member of ARCH or GARCH family models,
- ISD – option implied standard deviation *(please note, that we are not using any of such models in this thesis)*,
- SV – stochastic volatility forecasting *(please note, that we are not using any of such models in this thesis).*

In those papers, under their review, where models from both HISVOL and GARCH groups were studied, in 22 cases models from HISVOL group performed more accurate forecasts and in 17 papers models from GARCH group provided more accurate forecasts. For example, Boudoukh *et al.* (1997) comes to a conclusion that EWMA model provides a more accurate forecast (1-day forecast) than GARCH (1,1)[3]. In addition, McMillan *et al.* (2000) ranks forecasts from random walk and moving average models higher than ones from

[3] Boudoukh *et al.* (1997) used 'Realized volatility' as daily squared price changes averaged over 5 days.

GARCH models, using 1 day, 1 week and 1 month ahead forecasts, but should be remarked that underlying volatility proxies were squared returns (1 day, 1 week and 1 month returns respectively), so forecasts were one step ahead. On the other hand, Cumby *et al.* (1993) rank EGARCH model forecast higher than naive one, but in this case forecast horizon is one week. Hans Franses & van Dijk (1996) conclude that random walk model is better than GARCH but QGARCH (Quadratic GARCH or non-linear GARCH) is slightly better than random walk. Hans Franses & van Dijk (1996), used weekly squared deviations to approximate volatility.

As you can see from the previous paragraph, there is no clear view on this matter, whether to stick to simple models in volatility forecasting or to apply sophisticated GARCH-family models. Model selection is dependent on the length of the forecast horizon, current volatility regime and other factors, we provide forecasting results of different models, underlying assets[4] and horizons. Furthermore, results (namely, ranking based on forecast accuracy) may change if another forecasting accuracy metrics would be used, e.g. RMSE or MAE. Nevertheless, within GARCH-family models, it is clear that GARCH models are better than ARCH ones. Hansen & Lunde (2005) compared more than 300 GARCH-family models and concluded that simple ARCH model is clearly outperformed by GARCH extensions. Another important conclusion of Hansen & Lunde (2005) was that GARCH (1,1) is not outperformed by GARCH extensions. Nonetheless, Hansen & Lunde (2005) recognize that it may be due to limitations of their analysis (models were tested on two assets only: DM/$ exchange rate data and IBM stock prices data).

2.2 Literature on unconventional volatility forecasting

With the rise of computational power and data availability, machine learning techniques become more and more popular. Nowadays, it is one of the hottest topic for a research. Machine learning techniques have a wide range of applications: from speech recognition and image identification to customer classification and time series forecasting. We are considering machine learning techniques that are applicable to any time-series forecasting.

[4] Though we use only one class of assets, namely stocks, however from different sectors and of different sizes in term of market capitalization.

Connor *et al.* (1994) suggest that neural networks (NNs) for time-series are a special case of nonlinear auto-regressive models. In NNs for time-series, lags of a time-series are used as inputs for a neural network and output serves as a forecast. Forecast errors are then used for updating the weights in the network. Crone & Kourentzes (2010) introduced automatic methodology for specifying multilayer perceptions (MLP) that was ranked 2nd in the 2008 ESTSP[5] forecasting competition. A clear advantage of their methodology is that it is fully data driven and does not require any subjective assessment. In Kourentzes & Crone (2010)[6], they additionally provide a solution for automatic specification of inputs in NNs for forecasting. Authors use an Iterative Neural Filter for automatic frequency identification and feature extraction.

Adya & Collopy (1998) reviewed 48 papers on NNs forecasting. Though through their 'filters'[7] only 22 studies passed, NNs outperformed other forecasting techniques in 19 (86%) of them. In addition, artificial neural networks (ANNs) can show superior performance in cases when outliers in data are present. Barrow & Kourentzes (2018) compare NNs with conventional time-series forecasting methods on data with significant and seasonal outliers. They come to a conclusion that ANNs provide most accurate forecasts.

Donaldson & Kamstra (1997) introduced a semi-parametric nonlinear model based on symbiosis of GARCH model and neural networks (ANN-GARCH model). They came to a conclusion that ANN-GARCH model captures asymmetric and clustering properties of volatility better than GARCH, EGARCH or GJR-GARCH models. Therefore, ANN-GARCH out-of-sample forecasts take in account volatility properties better than other models.

De Stefani *et al.* (2017) statistically compare relationships between most used volatility proxies and then use Nonlinear Autoregressive with eXogenous (NARX) approach to forecast one volatility proxy using another volatility proxies as external regressors. De Stefani *et al.* (2017) use daily prices data for approximating volatility. They compare naive, GARCH(1,1), NAR, NARX, k Nearest Neighbors (kNN) and Support Vector Regression methods (SVR). They came to a conclusion that SVR model produces smaller forecast errors than those based on ANN and kNN. Moreover, all machine learning techniques outperform GARCH(1,1) model.

[5] European Symposium on Times Series Prediction.

[6] Note, that Crone & Kourentzes (2010) and Kourentzes & Crone (2010) are two different papers.

[7] Authors draw down criteria for effectiveness of validation and implementation. For more details see Adya & Collopy (1998).

Martinez *et al.* (2017) utilize kNN for time series forecasting. They develop an automatic method for parameters selection and additionally implement forecast combination in order to obtain more efficient results on NN3[8] competition data. kNN models are claimed to be computationally efficient. Likewise, Martinez *et al.* (2017) find that data deseasonalizing is not necessary since kNN seems to deal with it on its own. Nevertheless, detrending of the underlying time series is required.

2.3 Literature on forecast combination

From the above sections, we can see that there is no clear conclusion which models perform best in volatility forecasting or moreover different forecasting accuracy measures may provide no clear winner, for example: Hemanth Kumar & Basavaraj Patil (2015) results show that based on MAE – ARIMA forecast has the best performance and on contrary when results are ranked according to RMSE – forecast with feed-forward neural network provides the best result. Furthermore, there are papers suggesting that combined forecasts provide more accurate results. Forecast combination was popularized by Bates & Granger (1969) where they showed that a linear combination of two forecasts result in a smaller mean square error than either of two forecasts. However, they point out that if one forecast is already an optimal forecast (having minimum mean square error), then its weight in forecast combination will be 1 and consequently it will not be a combination of forecasts.

Makridakis & Winkler (1983) show that accuracy of a combined forecast rises with increasing number of forecasts being combined, but they also note that a marginal increase in forecast accuracy diminishes after combining more than 5 methods. Makridakis & Winkler (1983) utilized averaging technique for forecast combinations. In Winkler & Makridakis (1983)[9], authors utilize weighted average method instead of simple average as in Makridakis & Winkler (1983). Authors test 5 different techniques for weights determination that were introduced by Newbold & Granger (1974). Winkler & Makridakis (1983) conclude that 2 out of 5 weighting methods outperform other and also outperform individual forecasts. Likewise, they show that forecasts combined using weighted average have better performance than the ones using simple average.

[8] Artificial Neural Network & Computational Intelligence Forecasting Competition.

[9] Note, that Makridakis & Winkler (1983) and Winkler & Makridakis (1983) are two different papers.

Granger & Ramanathan (1984), introduced OLS based forecast combination technique, where individual forecasts are used as regressors. This technique performs especially well if individual forecasts are biased. Granger & Ramanathan (1984) show that in case individual forecasts are biased, OLS combined forecast provides better results than forecast combined using Bates & Granger (1969) minimum variance method. Likewise, there are 'extensions' of OLS combination methods that in fact are applications of some restrictions on coefficients of OLS combinations. For instance, Aksu & Gunter (1992) introduced Equality Restricted Least Squares (ERLS) combinations, where sum of all OLS coefficients is constrained to be equal to one. Therefore, we explicitly get weights for each forecasting method we combine. In addition, they introduce Non-negativity Restricted Least Squares (NRLS) combinations, where OLS coefficients are constrained to be non-negative. Aksu & Gunter (1992) conclude that performance of NRLS combined forecasts in most cases superior to ones from OLS and ERLS combined forecasts, and ERLS[10] combined forecasts outperform OLS combined forecasts in majority of cases. Diebold (1988) highlights that auto-correlation of residuals in OLS forecast combination model should not be disregarded. Residuals from OLS forecast combination model tend to be auto-correlated, thus it should be handled appropriately, since auto-correlation in residuals represents some hidden information that is still present in residuals. Diebold (1988) shows that ARMA adjustment to OLS combined forecast provides superior results.

Additionally, there is a method of neural networks combination that combines a set of neural networks (called ensemble) using various weighting methods (operators). Kourentzes *et al.* (2014) compare three ensemble operators: mode, median and mean. Authors draw a conclusion that the mode operator performs better than the median one and the median operator performs better than the mean one. Likewise, Kourentzes *et al.* (2014) claim that mode operator is most robust to distributional asymmetries in the forecasts of the members of an ensemble. In addition, Kourentzes *et al.* (2014) find that the mode operator is the most computationally efficient since it requires smaller amount of ensemble members to be trained.

Moreover, forecast combination can circumvent issue with selection of parameter k in kNN time series forecasting. Martinez *et al.* (2017) suggest fitting several models with different k's and then combining them. This method proved

[10] ERLS without an intercept.

to be more computationally efficient and accurate than usage of optimization tool for selection of k parameter for kNN time series forecasting.

It should be noted that in the above mentioned forecast combination methods authors combined forecasts created from more less same information sets (e.g. historical data). However, it is considered to be more beneficial when forecasts created using different information sets are being combined. Aiolfi *et al.* (2010) combine not only univariate time-series models but also forecasts from subjective surveys (such as Survey of Professional Forecasters), non-linear univariate and multivariate factor-augmented models. They find that in majority cases simple average of survey forecasts provide more accurate results than a forecast from the best univariate model but simple average combination of survey forecasts and forecasts from various time-series models performs relatively well. Nevertheless, there are examples when forecast combinations fail to outperform individual forecasts. For example Degiannakis (2018) shows that at forecast horizon of 5 trading days, combined forecasts (both simple and weighted averaged) fail to outperform realized volatility forecast from heterogeneous auto-regressive model. Nonetheless, when the forecast horizon is increased to 10 trading days, combined forecasts provide better results than individual forecasts. Degiannakis (2018) used forecast combination technique based on minimum forecast error in previous period.

Chapter 3

Data

Our raw data are stock daily prices data from yahoo Finance[1] from January 2015 until January 2018. Stocks we use are presented in Table 3.1, it is sorted according to market capitalization. In our sample, we have ten 'mega' cap stock, one 'large' cap stock, six 'mid' cap stocks and three 'small' cap stocks[2]. The sample is comprised of stocks from eight different sectors[3]. Stocks selection was random, though with one condition — availability of data.

Once we have our sample, we split it in two sub-samples: training and testing[4]. Testing sub-sample starts from January 1, 2018, end is dependent on a forecast horizon which is either 1 or 2 weeks (5 or 10 business days). Length of training sub-samples vary. There are two training sub-samples:

- from January 2015 until end-December 2017 (750+ observations);
- from January 2017 until end-December 2017 (245+ observations).

Different training sub-samples are needed in order to test hypotheses #2 and #3.

Once we have both in-sample and out-of-sample, we calculate different proxies for volatility. One of the challenges in volatility forecast is what to use as a proxy for volatility (standard deviation, squared return etc.). If it is a standard deviation, then what period for returns should be used in order to calculate it. Since we are going to model daily volatility and our sampling frequency is

[1] We are using getSymbols function from quantmod package in R (Ryan & Ulrich (2019)) that downloads daily highest, lowest, closing and opening prices as well as trading volume and adjusted prices (the latter two are not used in our calculations or manipulations).

[2] Stock capitalization segmentation: 'mega' – more than $200 billion, 'large' – more than $10 billion, 'mid' – more than $2 billion, 'small' – more than $300 million.

[3] Sectors are defined according to Yahoo Finance.

[4] Can also be referred to as in-sample and out-of-sample.

Table 3.1: Stocks used for calculation

Stock	**Ticker**	**Sector**	**Market Cap ($B)**	**Cap type**
Microsoft Corporation	MSFT	Technology	1051	Mega
Apple Inc.	AAPL	Technology	939.457	Mega
Alphabet Inc.	GOOG	Technology	796.361	Mega
Facebook, Inc.	FB	Technology	578.006	Mega
JPMorgan Chase & Co.	JPM	Fin Services	370.461	Mega
Johnson & Johnson	JNJ	Healthcare	350.747	Mega
Exxon Mobil Corporation	XOM	Energy	320.121	Mega
Mastercard Incorporated	MA	Fin Services	281.238	Mega
The Walt Disney Company	DIS	Consumer C.	257.128	Mega
The Coca-Cola Company	KO	Consumer D.	222.35	Mega
Splunk Inc.	SPLK	Technology	20.866	Large
Teva Pharmaceutical Industries Limited	TEVA	Healthcare	9.155	Mid
Chegg, Inc.	CHGG	Consumer D.	5.204	Mid
GW Pharmaceuticals plc	GWPH	Healthcare	5.023	Mid
Manchester United plc	MANU	Consumer C.	2.978	Mid
Chesapeake Energy Corporation	CHK	Energy	2.785	Mid
United States Steel Corporation	X	Basic Materials	2.598	Mid
Scientific Games Corporation	SGMS	Consumer C.	1.782	Small
Ship Finance International Limited	SFL	Industrials	1.53	Small
AAR Corp.	AIR	Industrials	1.477	Small

Consumer C. – Consumer Cyclical; Consumer D. – Consumer Defensive.

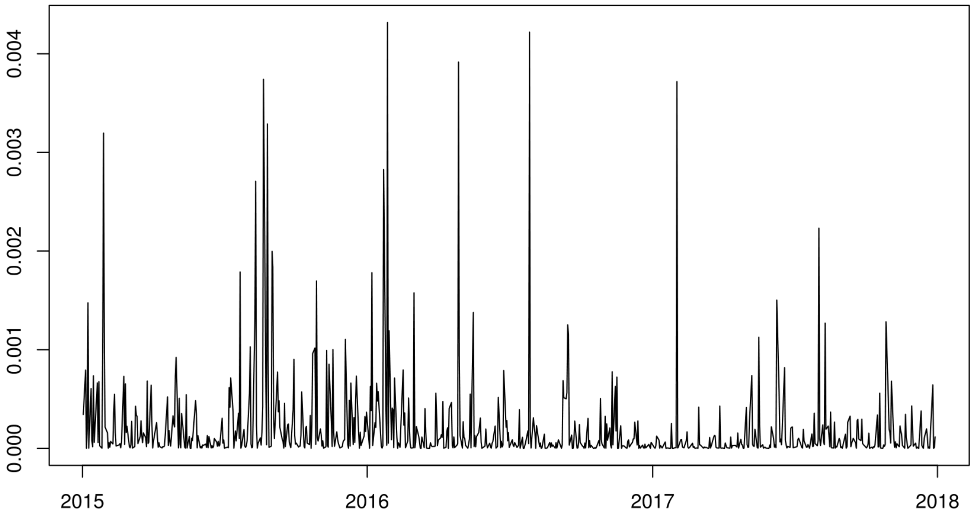
also daily, we cannot use usual standard deviation. Standard deviation can be defined as a square root of sample variance:

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{t=1}^{N}(r_t - \bar{r})^2, \tag{3.1}$$

In daily data, N from equation Equation 3.1 equals to one or four, which is still insufficient. So we utilize volatility proxies that allow us to use available data at daily sampling frequency. In total, we have 4 proxies. Below we present their formal definitions and visual examples for AAPL stock:

1. Squared return volatility proxy: visual example is presented in Figure 3.1. Definition is: $\hat{\sigma}_t = r_t^2$.

Figure 3.1: Jan 2015 – Dec 2017 AAPL squared return volatility proxy.



Source: Authors calculations.

2. Absolute return volatility proxy: visual example is presented in Figure 3.2. Formal definition is: $\hat{\sigma_t} = |r_t|$. In both absolute return and squared return proxies, r_t is a simple daily return that takes closing stock prices, C_t, as input and defined as:

$$r_t = \frac{C_t - C_{t-1}}{C_{t-1}}$$

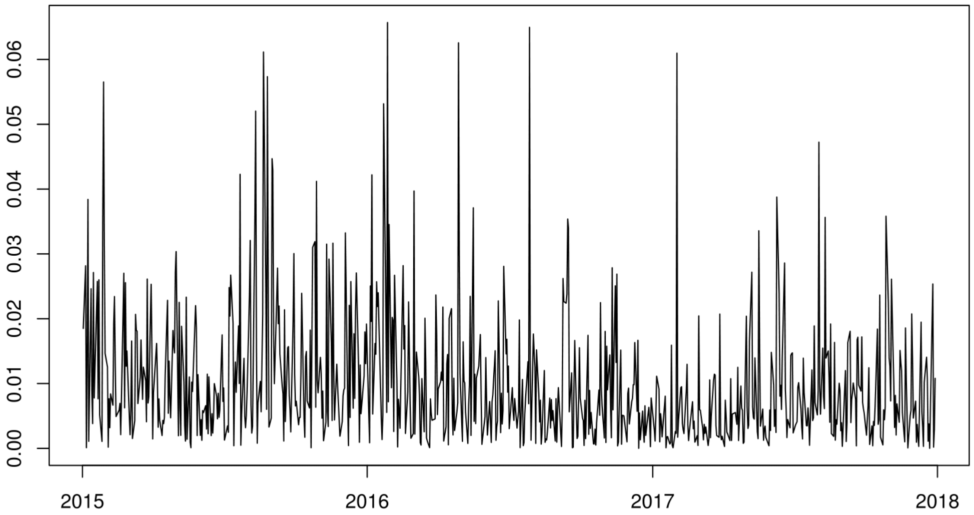
3. High-low Parkinson proxy: visual example is presented in Figure 3.3. Parkinson estimate can be formally defined as:

$$\hat{\sigma_t}^2 = \frac{(\ln H_t - \ln L_t)^2}{4 \ln 2} \tag{3.2}$$

4. Garman-Klass proxy: visual example is presented in Figure 3.4. Formal definition is presented below:

$$\hat{\sigma_t}^2 = 0.5(\ln H_t - \ln L_t)^2 - 0.39(\ln C_t - \ln O_t)^2 \tag{3.3}$$

Figure 3.2: Jan 2015 – Dec 2017 AAPL absolute return volatility proxy.



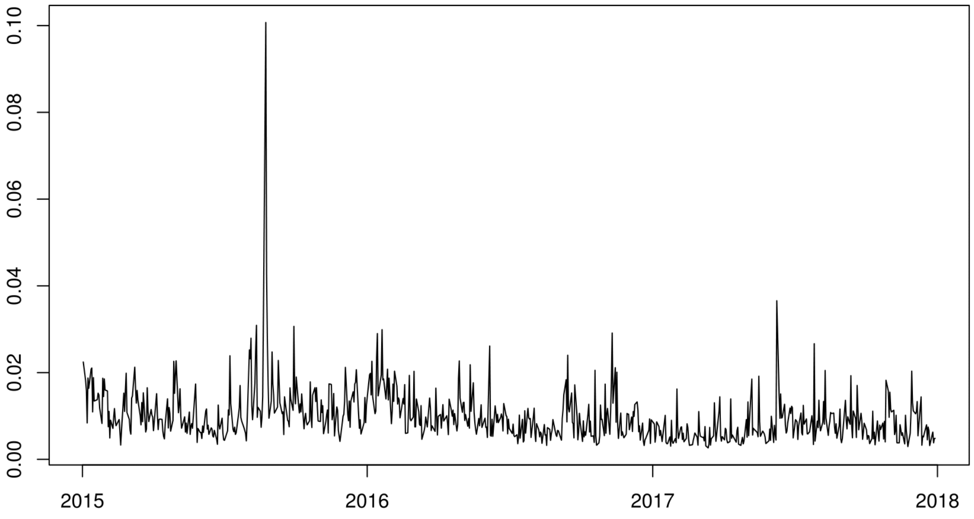
Source: Authors calculations.

In Equation 3.2 and Equation 3.3, H_t, L_t, C_t and O_t are highest, lowest, closing and opening prices at time t, respectively.

From formal definitions, it is seen that squared returns and absolute returns proxies use returns as inputs while Parkinson and Garman-Klass proxies utilize prices data as inputs. Visual representation example of inputs for squared returns and absolute returns proxies (i.e. daily returns) is displayed in the left plot of Figure 3.6. The right plot of Figure 3.6 shows returns histogram, it is seen that returns have leptokurtic distribution, that is: presence of fat tails and high density around zero, and how it is different from normal distribution (solid black line). Visual representation example of inputs for Parkinson and Garman-Klass volatility proxies is displayed in Figure 3.7, we plot all inputs (Opening, Closing, Highest and Lowest daily prices) without any distinction since they are very close to each other and there is no value added from it. However, you may see that there is some dispersion between them and they are not exactly the same.

Additionally, as you may see from visual representations, absolute return volatility proxy is the noisiest out of all four volatility proxies. Additionally, it is seen, that Parkinson proxy and Garman-Klass proxy are similar and both

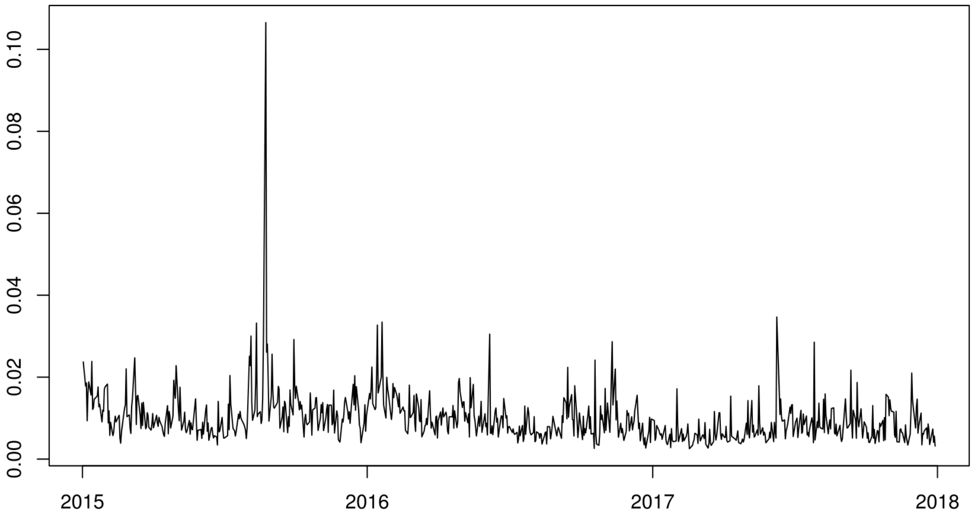
Figure 3.3: Jan 2015 – Dec 2017 AAPL Parkinson volatility proxy.



Source: Authors calculations.

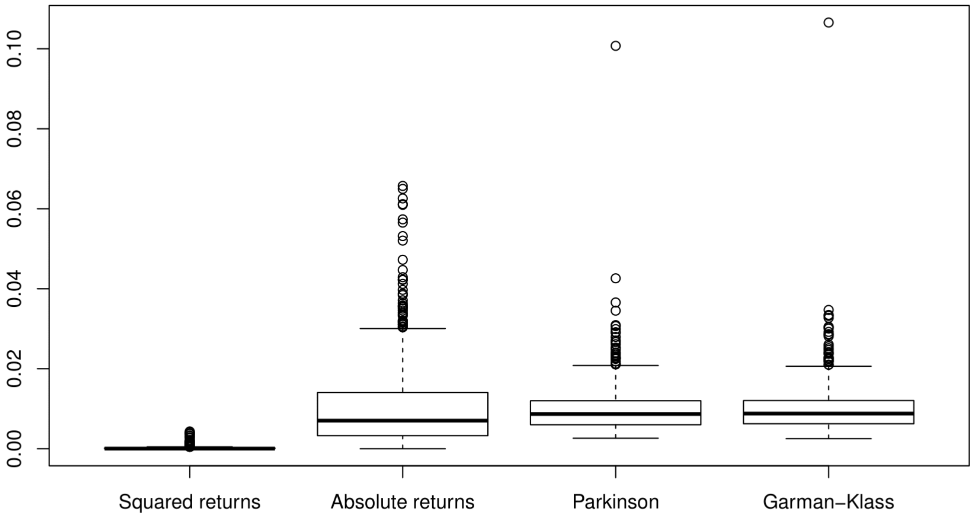
are the least noisy volatility proxies. Likewise, both proxies have a spike in the second half of 2015, which is clearly an outlier. On the other hand, squared return volatility proxy and absolute return volatility proxy do not have such clear outliers but they are much noisier. From visual assessment, we may see that noisier the proxy, harder it to detect outliers. That is, in absolute return volatility proxy it is harder to distinguish outliers from noise than in squared return volatility proxy and in squared return volatility proxy it is harder to distinguish outliers from noise than in Parkinson or Garman-Klass volatility proxies. From Figure 3.5, it is seen that squared returns volatility proxy has the smallest range among all four proxies, whereas Parkinson and Garman-Klass proxies have very similar ranges and absolute returns proxy has the biggest range (excluding outliers).

Figure 3.4: Jan 2015 – Dec 2017 AAPL Garman-Klass volatility proxy.



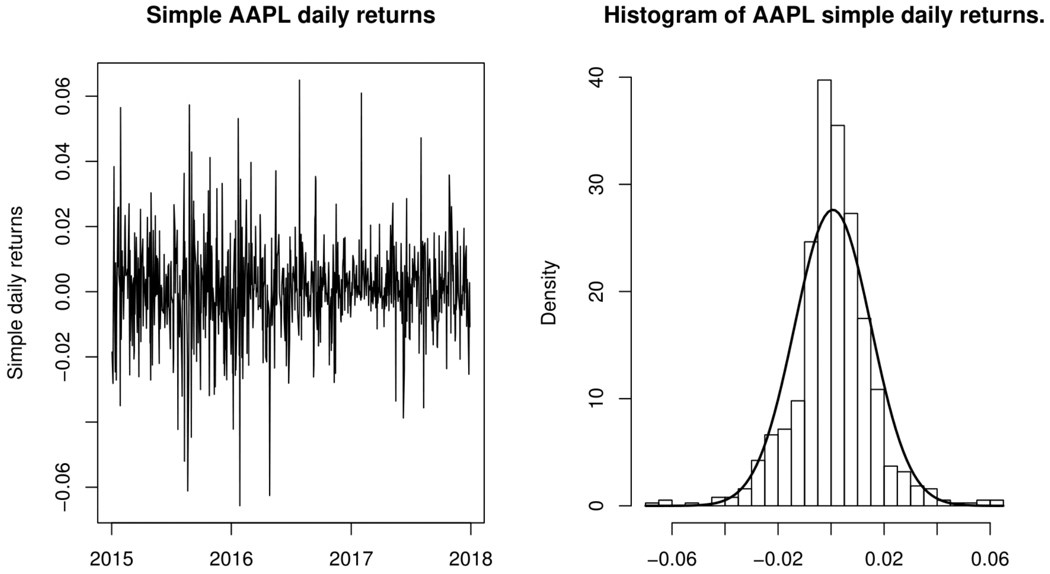
Source: Authors calculations.

Figure 3.5: Boxplots of volatility proxies for AAPL (Jan 2015 – Dec 2017).



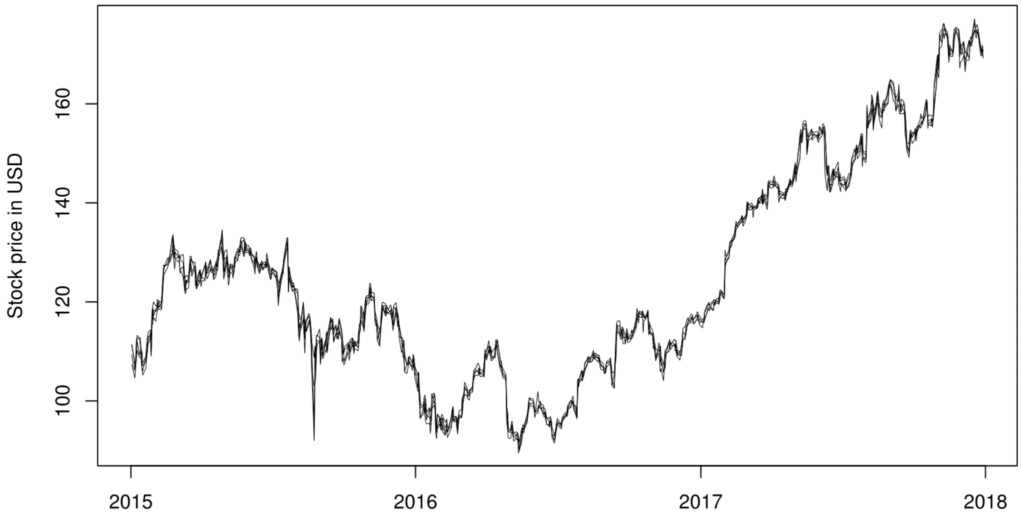
Source: Authors calculations.

Figure 3.6: Jan 2015 – Dec 2017 AAPL simple daily returns.



Source: Authors calculations.

Figure 3.7: Jan 2015 – Dec 2017 AAPL prices.



Source: Yahoo Finance.

Chapter 4

Methodology

This chapter lists and describes methods we use for our calculations. Motivation to use them is driven by discussion in Chapter 2, therefore, in this chapter, we do not dive deep in discussions of each method's advantages or drawbacks since main purpose of this chapter is to present what methods we use to a reader. This chapter is segmented in four parts:

1. Traditional time-series models – where we introduce traditional time-series models we use for forecasting and subsequent combination.

2. Unconventional time-series models – where we introduce unconventional time-series models we use for forecasting and subsequent combination.

3. Forecasts combination – where we introduce forecasts combination methods we use.

4. Forecasts evaluation and comparison – where we describe methods we use for forecasts performance evaluation and how we compare them.

4.1 Traditional time-series models

In this section, we present traditional time-series models we use. This section is split in three subsections:

1. Simple models,

2. ARIMA models,

3. GARCH models.

4.1.1 Simple models

Below, we present simple models that do not require much of data manipulation and therefore are easy to implement.

- Historical average,
- Simple Moving Average (SMA),
- Adjusted EWMA.

Historical average is a time-invariant simple mean over training sample. Forecast is also a constant and equal to a simple mean of in-sample data.

$$\hat{\sigma} = \frac{\sum_{i=1}^{T} \sigma_i}{T} \tag{4.1}$$

SMA is a simple filter. As an input it takes a signal and calculates a simple average for n-period window. In our calculations we use a window of length 10. The forecast is calculated as:

$$\hat{\sigma}_{t+k} = \frac{\sum_{j=t-n+k}^{t+k-1} \sigma_j}{n} \tag{4.2}$$

The above reflects a simple average over last n observations. When we do an out-of-sample forecast further than one step (i.e. $k > 1$), we use a forecast for the first step as one of inputs in the filter[1]. That is, only one step ahead forecast explicitly utilizes actual data. At step $t + k$, where $k = n$, SMA filters already filtered data. Furthermore, estimate of $\hat{\sigma}_{t+k}$, where $k >> n$, converges to some constant. Therefore, such method of forecasting cannot be used for a long-term forecasting since quality of such a forecast deteriorates significantly once $k > n$ since model starts to filter already filtered data.

Adjusted EWMA is in fact auto-regressive model that has a smoothing function with more weight given to recent observations. Since we need an out-of-sample forecasting, we adjust ordinary EWMA so it can produce a purely out-of-sample forecast (i.e. not to feed the model with actual observations). Adjusted EWMA is defined as:

$$\hat{\sigma}_{t+k} = (1 - \lambda)\sigma_{t+k-2} + \lambda\sigma_{t+k-1}, \tag{4.3}$$

[1] $\hat{\sigma}_{t+1}$ is used in calculation of $\hat{\sigma}_{t+2}$.

where $\lambda \in [0,1]$ is a decay constant. Greater λ – stronger the influence of volatility from previous period. Second term in the above equation is the persistence term that determines how volatility from the previous period is carried over to the current period notwithstanding what happened in the current period. λ closer to 1 would result in a very persistent volatility estimator. For daily data, popular value of λ is 0.94 that was recommended by RiskMetrics (1996). Nonetheless, Bollen (2015) argues Riskmetrics value and recommends to set λ to 0.72 (when RMSE and MAE criteria are used), however, for data with heteroscedasticity, which financial data usually are, λ equal to 0.88 results in lower heteroscedasticity adjusted RMSE and heteroscedasticity adjusted MAE. For $k > 2$, the model starts to utilize already filtered data, thus a forecast from adjusted EWMA converges to some constant faster than a forecast from SMA whenever $n > 2$ from Equation 4.2.

4.1.2 AR(F)IMA models

ARIMA is an extension of ARMA model. ARMA model describes time series (in our case volatility) in terms of two polynomials: auto-regression and moving average of errors:

$$\hat{\sigma_t} = c + \varepsilon_t + \sum_{i=1}^{p} \phi \sigma_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}, \tag{4.4}$$

where $|\sum \phi| < 1$ and $|\sum \theta| < 1$ for stationary process, p is the order of AR and q is the order of MA. Errors ε_t assumed to be i.i.d.

Since our time-series can be non-stationary, we may first-difference it in order to obtain stationary process. Differencing is the discrete-time version of differentiation. ARIMA models are ARMA models with differenced input, i.e. the underlying time-series has been differenced:

$$\Delta y_t = y_t - y_{t-1}, \tag{4.5}$$

in this case Δy_t is differenced time-series. Equation 4.4 can be rewritten using back-shift operator, which is defined as:

$$B^k X_t = X_{t-k}, \tag{4.6}$$

where X_t is any time-series. Then, Equation 4.4 can be expressed as:

$$\phi(B)\sigma_t = \theta(B)\varepsilon_t, \tag{4.7}$$

where polynomials are:

$$\phi(z) = 1 - \phi_1 z - ... - \phi_p z^p$$

and

$$\theta(z) = 1 - \theta_1 z - ... - \theta_p z^q$$

We introduced the back-shift operator in order to express ARIMA process in a more compact form. Equation 4.5 can be expressed with the use of back-shift operator as:

$$(1 - B)^d y_t, \tag{4.8}$$

where $d \in \mathbb{Z}$ is the level of differencing. Generally, $d = 1$ because usually it is enough for a non-stationary time series to be differenced once in order to become stationary. ARIMA can be defined as:

$$\phi(1 - B)^d y_t = \theta(B)\varepsilon_t \tag{4.9}$$

Whenever, d is fractional, i.e. not an integer, we obtain a long memory process that is explained by ARFIMA model[2]. The model formulated in same way as in Equation 4.9 but $d \in (0, 1)$.

Order selection for ARMA, models can be done using Box & Jenkins (1970) method. Box-Jenkins method consists of 3 steps:

1. Model identification: identification of model orders (AR and MA), assurance of stationarity, determination of seasonality;

2. Parameter estimation: estimate parameters so ARIMA model is best fit for a given process;

3. Model check: confirmation of i.i.d. residuals.

AR and MA order identification (parameters p and q from Equation 4.4) can be done with the use of auto-correlation function (ACF) and partial auto-

[2] ARFIMA model is implemented as in Ghalanos (2019).

correlation function (PACF). ACF is computed as:

$$\rho_s = \frac{\gamma_s}{\gamma_0} = \frac{Cov(x_t, x_{t-s})}{Var(x_t)} \tag{4.10}$$

PACF captures correlation between two lags that is not explained by correlations at all lower-order-lags.

In AR (p) model, PACF of lag p should not be zero, but all subsequent partial auto-correlations should be insignificant. In MA (q) model, ACF of lag q should be significant and all subsequent auto-correlations should be close to zero. In ARMA (p, q) model, ACF and PACF are not very informative in order identification. For this purpose, information criteria can be used. The information criteria vary according to the penalty term. The most popular information criterion is Akaike (AIC), which is almost always provided by standard statistical software. Hyndman & Athanasopoulos (2018) define AIC as:

$$\text{AIC} = T \log\left(\frac{\text{SSE}}{T}\right) + 2(k+2), \tag{4.11}$$

Where T is the number of observations used for estimation and k is the number of predictors in the model, SSE is a fit of the model. There exists an extension of AIC — AIC with a correction (AICc). Hyndman & Athanasopoulos (2018) define AICc as:

$$\text{AIC}_\text{c} = \text{AIC} + \frac{2(k+2)(k+3)}{T-k-3} \tag{4.12}$$

Basically, AICc is AIC with an extra penalty term for the number of parameters. However, with increasing sample size, extra penalty term converges to zero and subsequently AICc converges to AIC.

Likewise, Schwarz's Bayesian information criterion (SBIC or BIC) is widely used. Hyndman & Athanasopoulos (2018) define BIC as:

$$\text{BIC} = T \log\left(\frac{\text{SSE}}{T}\right) + (k+2)\log(T) \tag{4.13}$$

Model with the lowest information crietion is selected. In our model selection, we are using the Hyndman & Khandakar (2008) algorithm that combines unit root tests, minimization of AICc and MLE to get ARIMA orders. The Hyndman-Khandakar algorithm description from Hyndman & Athanasopoulos (2018, Section 8.7):

1. The number of differences $0 \leq d \leq 2$ is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimising the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space..
   (a) Four initial models are fitted:
      - ARIMA(0,d,0),
      - ARIMA(2,d,2),
      - ARIMA(1,d,0),
      - ARIMA(0,d,1).

      A constant is included unless $d = 2$. If $d \leq 1$, an additional model is also fitted:
      - ARIMA(0,d,0) without a constant.

   (b) The best model (with the smallest AICc value) fitted in step (a) is set to be the ”current model”.

   (c) Variations on the current model are considered:
      - vary p and/or q from the current model by ±1;
      - include/exclude c (a constant) from the current model.

      The best model considered so far (either the current model or one of these variations) becomes the new current model.

   (d) Repeat Step 2(c) until no lower AICc can be found.

4.1.3 GARCH models

We are implementing GARCH models using ’rugarch’ R package by Ghalanos (2019). GARCH definitions are taken from Ghalanos guide to the package (Ghalanos (2017)), though the guide is for an older version of the package, it is still relevant. In our models fitting, we are setting GARCH orders to $p = 1$ and $q = 1$. Additionally, we are estimating the intercept and do not utilize variance-targeting. Likewise, we are not adding any external regressors. Finally, the conditional density used for innovations is skewed Student-T distribution since it is more suitable for financial data than normal Gaussian distribution. Major advantage of Student-T distribution is that it accounts for excess kurtosis and can reflect leptokurtic property of financial data. We are using skewed version of Student-T distribution to account for any possible skewness, so in case there is no skew, skew parameter will equal to zero and usual Student-T distribution will be assumed. Gao *et al.* (2012) conclude that GARCH model with Student-T distribution performs better than the one with

Normal distribution assumption[3]. GARCH models we utilize for our forecast combination are presented below:

1. Standard GARCH,
2. AVGARCH,
3. GJR GARCH,
4. TGARCH,
5. NARCH,
6. NAGARCH,
7. APARCH,
8. ALLGARCH.

Standard GARCH model of Bollerslev (1986) is defined as:

$$\hat{\sigma}_t^2 = \omega + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{4.14}$$

Sum of estimates α and β can be interpreted as persistence parameter $(\hat{P})$, which is a quantification of GARCH's ability to capture volatility clustering. The unconditional variance of the GARCH model, $\hat{\sigma}^2$, can be calculated as follows:

$$\hat{\sigma}^2 = \frac{\hat{\omega}}{1 - \hat{P}} \tag{4.15}$$

AVGARCH — Absolute Value GARCH model of Taylor (1986) and Schwert (1990) is defined as:

$$\sigma_t = \omega + \sum_{j=1}^{q} \alpha_j |\varepsilon_{t-j}| + \sum_{j=1}^{p} \beta_j \sigma_{t-j} \tag{4.16}$$

In fact, this is a standard GARCH model, where squared values are substituted with absolute values, note that σ_{t-j} does not need to be absolute since it is

[3] Worth mentioning that Gao *et al.* (2012) also conclude that GARCH with Generalized Error distribution performs better than GARCH with Student-T distribution.

always non-negative. The persistence parameter is defined as:

$$\hat{P} = \sum_{j-1}^{q} \alpha_j \kappa_j + \sum_{j-1}^{p} \beta_j,$$

where κ_j is expectation of the standardized residuals z_t and defined as:

$$\kappa_j = E(|z|) = \int_{-\infty}^{\infty} |z| f(z, 0, 1, ...) dz$$

Unconditional variance of the AVGARCH model, $\hat{\sigma}^2$, is calculated as squared value of Equation 4.15.

GJR GARCH — GARCH extension of Glosten, Jagannathan and Runkle is defined as:

$$\sigma_t^2 = \omega + \sum_{j=1}^{q} (\alpha_j \varepsilon_{t-j}^2 + \gamma_j I_{t-j} \varepsilon_{t-j}^2) + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{4.17}$$

where γ_j represents leverage term, an indicator function I that equals 1 for $\varepsilon \leq 0$ and 0 otherwise. GJR GARCH model of Glosten *et al.* (1993) models positive and negative shocks on the conditional variance asymmetrically via the use of the indicator function I. Therefore, the persistence term is augmented and defined as:

$$\hat{P} = \sum_{j-1}^{q} \alpha_j + \sum_{j-1}^{p} \beta_j + \sum_{j-1}^{q} \gamma_j \kappa,$$

where κ is the probability of standardized residuals z_t being below zero and calculated as:

$$\kappa = E[I_{t-j} z_{t-j}^2] = \int_{-\infty}^{0} f(z, 0, 1, ...) dz,$$

where $f(...)$ is a standardized conditional density. Whenever distribution is symmetric, $\kappa = 0.5$. Unconditional variance is calculated as in Equation 4.15.

TGARCH — Threshold GARCH model of Zakoian (1994) is defined as:

$$\sigma_t = \omega + \sum_{j=1}^{q} \alpha_j (|\varepsilon_{t-j}| - \gamma_j \varepsilon_{t-j}) + \sum_{j=1}^{p} \beta_j \sigma_{t-j} \tag{4.18}$$

The model is very similar to AVGARCH, only with an addition of a leverage term (γ). Persistence term is defined similarly as in AVGARCH, however, κ_j

is defined a bit differently:

$$\kappa_j = E(|z| - \gamma_j z) = \int_{-\infty}^{\infty} (|z| - \gamma_j z) f(z, 0, 1, ...) dz$$

Unconditional variance is calculated in the same way as in AVGARCH.

NARCH — Nonlinear ARCH model of Higgins & Bera (1992) is defined as:

$$\sigma_t^\lambda = \omega + \sum_{j=1}^{q} \alpha_j |\varepsilon_{t-j}|^\lambda \tag{4.19}$$

Persistence term is defined similarly as in AVGARCH, but definition of κ_j differs:

$$\kappa_j = E(|z_{t-j}|)^\lambda = \int_{-\infty}^{\infty} |z|^\lambda f(z, 0, 1, ...) dz$$

Unconditional variance is calculated similarly to the one of standard GARCH:

$$\hat{\sigma}^2 = \left(\frac{\hat{\omega}}{1 - \hat{P}} \right)^{2/\lambda}$$

NAGARCH — Nonlinear Assymetric GARCH model of Engle & Ng (1993) is defined as:

$$\sigma_t^2 = \omega + \sum_{j=1}^{q} \alpha_j \sigma_{t-j}^2 (|z_{t-j} - \eta_{2j}|)^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2, \tag{4.20}$$

where η_{2j} is shift parameter of the absolute value function transformation. Persistence term is defined similarly as in AVGARCH, but definition of κ_j differs:

$$\kappa_j = E(|z_{t-j} - \eta_{2j}|)^2 = \int_{-\infty}^{\infty} |z - \eta_{2j}|^2 f(z, 0, 1, ...) dz$$

Unconditional variance is calculated as in Equation 4.15.

APARCH — Asymmetric Power ARCH model of Ding *et al.* (1993) is defined as:

$$\sigma_t^\delta = \omega + \sum_{j=1}^{q} \alpha_j (|\varepsilon_{t-j}| - \gamma_j \varepsilon_{t-j})^\delta + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^\delta, \tag{4.21}$$

where $\delta \in \mathbb{R}^+$ is a Box & Cox (1964) transformation of σ_t. APARCH captures the Taylor effect and leverage (γ). Taylor effect (Taylor (1986)) states that in majority cases the sample auto-correlation of absolute returns is greater than that of squared returns. Persistence term is defined similarly as in AVGARCH, but definition of κ_j differs and resembles the one from TGARCH with a minor difference:

$$\kappa_j = E(|z| - \gamma_j z)^{\delta} = \int_{-\infty}^{\infty} (|z| - \gamma_j z)^{\delta} f(z, 0, 1, ...) dz$$

Calculation of unconditional variance is same as in NARCH, but $\lambda = \delta$. In fact, GARCH, AVGARCH, GJR GARCH, TGARCH and NARCH are all sub-models of APARCH with parameters γ_j and δ assuming different values (in case of NARCH also $\beta_j = 0$).

ALLGARCH — Full Family GARCH model of Hentschel (1995) is defined as:

$$\sigma_t^{\lambda} = \omega + \sum_{j=1}^{q} \alpha_j \sigma_{t-j}^{\lambda} (|z_{t-j} - \eta_{2j}| - \eta_{1j}(z_{t-j} - \eta_{2j}))^{\lambda} + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^{\lambda}, \tag{4.22}$$

this is a Box-Cox transformation of σ_t where λ is its shape parameter, η_{1j} and η_{2j} are rotation and shift parameters of the absolute value function transformation respectively. Full Family GARCH model is a more general version of APARCH model[4]. It captures decomposition of the residuals, shifts and rotations of the news impact curve with shift being main source of asymmetry for smaller shocks and rotation capturing larger shocks. In fact, GARCH, AVGARCH, GJR GARCH, TGARCH, NARCH, EGARCH and APARCH are sub-models of Full Family GARCH model with additional restrictions on some parameters. Persistence parameter is defined similarly as in AVGARCH model, but κ_j has a more generalized formulation and defined as:

$$\kappa_j = E(|z_{t-j} - \eta_{2j}| - \eta_{1j}(z_{t-j} - \eta_{2j}))^{\lambda} = \int_{-\infty}^{\infty} (|z - \eta_{2j}| - \eta_{1j}(z - \eta_{2j}))^{\lambda} f(z, 0, 1, ...) dz$$

Unconditional variance is calculated in the same way as in NARCH[5].

4.2 Unconventional time-series models

This section presents unconventional time-series models we use. We call them unconventional since they are from machine learning field.

[4] APARCH is a sub-model of Full Family GARCH model with $\eta_{2j} = 0$ and $|\eta_{1j}| \leq 1$.

[5] In fact, it is the other way around: unconditional variance of NARCH is calculated in the same way as in ALLGARCH.

NNAR — is a neural network auto-regression model implemented by Hyndman *et al.* (2019). NNAR is a feed-forward neural network that uses lagged values as inputs. NNAR we use has one hidden layer. NNAR model has two arguments p and k, where p is the number of lagged inputs and k is the number of nodes in the hidden layer. NNAR(p,0) is equivalent to ARIMA(p,0,0) model. p is selected automatically and is same as for AR(p) based on AIC. Number of nodes in the hidden layer can be either set manually or (if not set manually) is calculated as:

$$k = \text{round}((p+1)/2)$$

An example of NNAR(4,3) is depicted in Figure 4.1. On Figure 4.1, we can see input-output scheme. Inputs to each hidden node are linearly weighted via the below function:

$$z_j = b_j + \sum_{i=1}^{N} w_{i,j} x_i,$$

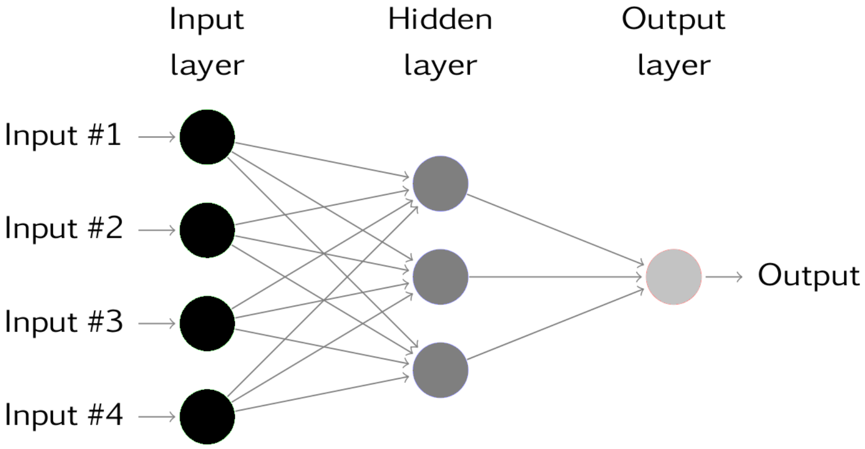
where N is the number of nodes in the hidden layer, x_i is the value of the node i from the input layer, $w_{i,j}$ is the weight of the input i from the input layer to the hidden node j, b_j is the intercept and z_j is the resulting input for the hidden node j. Afterwards, z_j is modified using a non-linear function (sigmoid) in order to produce an output from the hidden node j via the below funciton:

$$s(z) = \frac{1}{1+e^{-z}},$$

where $s(z)$ is the final output. At the initial iteration, weights $w_{i,j}$ take random values and then optimized at every iteration using underlying data in order to provide the best fit (or minimize the cost function that is mean square error). The intercept b_j is 'trained' in the same way. The network is trained t times and each time different random starting values for weights are taken, after t iterations, final result is the average of those t trained networks. In our application, we train our NNAR(p, k) models 20 times as suggested in Hyndman *et al.* (2019).

MLP — is a multilayer perception neural network for time-series forecasting. Major difference from NNAR(p, k) is option to have multiple hidden layers and different transfer function (NNAR utilizes sigmoid transfer function). In our implementation of MLP model, which is based on Kourentzes (2019), we use two hidden layers. MLP with one single hidden layer is similar to NNAR but transfer function can differ (instead of sigmoid function, hyperbolic tangent

Figure 4.1: A neural network with four inputs and one hidden layer with three nodes and denoted as NNAR(4,3).



Source: Hyndman & Athanasopoulos (2018).

function can be used). MLP can be defined as:

$$y_{t+1} = b_0 + \sum_{i=1}^{H} b_i g(a_{0,i} + \sum_{j=1}^{I} a_{j,i} x_j),$$

where transfer function $g(.)$ is a hyperbolic tangent activation function and defined as:

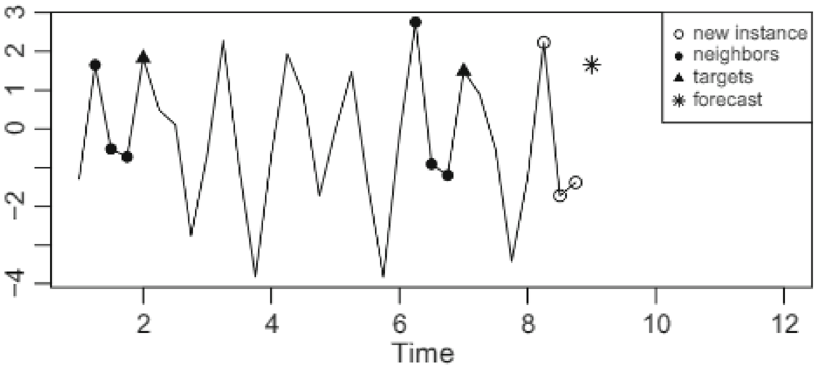
$$g(x) = \tanh(x) = \frac{2}{(1 - e^{2x}) - 1}$$

KNN regression — is a method of time series forecasting using k Nearest Neighbours regression technique. We implement KNN regression based on Martinez (2019) and Martinez *et al.* (2017). In Martinez *et al.* (2017, Section 2), k-NN time series forecasting is described as:

> ”Given some features – explanatory variables – of a new instance to be classified – regressed on – k-NN finds the k training instances that are closest to the new instance according to some distance metric and returns their majority class – average explained variable.”

In case of univariate time series forecasting, explanatory variables are lagged values of the time series and ”a new instance to be classified” is a forecast $t+h$. k-NN tries to identify any repetitive pattern in the data and utilize it in order to produce a forecast. On Figure 4.2 you can see an example of k-NN (with $k = 2$) one step ahead forecast with 3 lags used as regressors: new instances (empty dots) are regressed on two sets of the nearest neighbours (black dots) and their targets (triangles) are used to create a forecast (asterisk).

Figure 4.2: One-step-ahead forecast with 2-NN regression (i.e. $k = 2$).



Source: Martinez *et al.* (2017).

Before running k-NN forecast, we need to find optimal k. Martinez *et al.* (2017) describe three ways of choosing k:

1. Rule of thumb: $k = \text{round}(\sqrt{N})$, where N is the number of training instances.

2. Estimate optimal k — training set is split in training and validation parts and then optimal k is the one that minimizes forecasting error (selected forecasting accuracy measure can be selected) in validation set.

3. Combination of multiple models — several k-NN models (i.e. with different k's) produce forecasts and then it is averaged (mean, median or weighted mean can be used).

We utilize the third option since this requires less computational power than the second option and is more robust than the first option, moreover, it benefits from forecasts combination. Forecasts are combined using a median operator as Kourentzes *et al.* (2014) asserts it is more efficient and robust than a mean operator. For nearest neighbours selection, the Euclidean distance is used since it is considered to be a benchmark.

Likewise, lags should be selected, Martinez *et al.* (2017) present three alternatives:

1. Set number of lags equal to the period of seasonality, e.g. 1-12 lags for monthly data, lags 1-4 for quarterly data.

2. Analyze PACF and select lags with significant auto-correlations.

3. Wrapper approach — lags are selected based on predictive performance.

We utilize the second option, since the third one is computationally exhaustive and the first option is also computationally exhaustive for daily data (1-252 lags for daily data).

For multi-step ahead forecasting, Martinez *et al.* (2017) suggest three options:

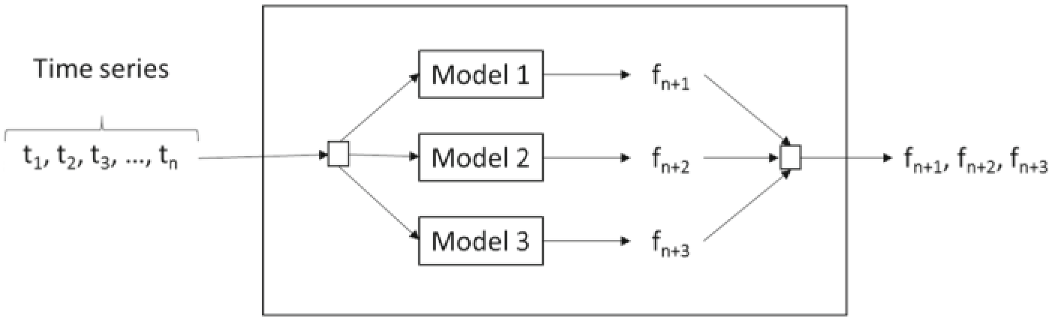
1. Recursive forecating – ARIMA, EWMA, etc. use this principle for multi-step ahead forecasting. Under this principle, forecast points are used as inputs for further forecast.

2. Direct approach – for each point in future a separate independent forecast is created. Under this approach only historical data is used to create the forecast, i.e. no forecast points are used.

3. Multi-Input Multi-Out (MIMO) – forecasts all future points at once.

In our application, we utilize MIMO strategy, since recursive forecasting may have a large impact from cumulative forecast errors and at some point forecast may converge to some constant (as in ARIMA and EWMA for a long forecast horizon). Direct approach is quite computationally extensive since for each point in future a separate model should be estimated. For example, for three-steps ahead forecasting, three separate models are created (see Figure 4.3 illustrative example). Additionally, Ben Taieb *et al.* (2012) in their comparison of the above methods for multi-step forecasting find that MIMO strategy provide the most accurate results.

Figure 4.3: The direct approach for three-steps ahead forecasting.



Source: Martinez *et al.* (2017).

4.3 Forecasts combination

We implement the following combination methods:

- Simple mean method,
- OLS combination,
- ARIMA on combined fit,
- NNAR on combined fit,
- KNN on combined fit.

Simple mean method is the easiest combination method and many researches claim it to be very efficient (e.g. Makridakis & Winkler (1983), Aiolfi *et al.* (2010)). The forecast for period $t+f$ of simple mean method is a simple average of forecasts, formally can be defined as:

$$y_{t+f} = \frac{\sum_{i=1}^{N} x_{i,t+f}}{N} \tag{4.23}$$

where N is the number of forecasts to be combined, $x_{i,t+f}$ is a forecast for period $t+f$ from forecasting method i, $f \in [1, 2, ...H]$ and H is a forecast horizon. This method can be prone to outliers but if many forecasts are combined, negative effect of outliers is lower. A clear advantage of this forecast is that it is very simple and straightforward to implement.

OLS combination was introduced by Granger & Ramanathan (1984). We use this combination technique as a standalone method and also as an intermediate step for other 3 methods that we describe later below. For OLS combination, the following steps are carried out:

1. We create our sample for OLS from fitting all standalone forecast methods on our in-sample data.[6]
2. We fit OLS without the intercept on our in-sample data, we fit it without the intercept since in our in-sample data we already have one constant, which is mean forecast. Additionally, from this step, we obtain a combined fit (i.e. fitted values of this OLS regression).

[6] Since there are eight standalone models from GARCH family, we decided to combine them in one via a separate OLS fit and then use a result of this OLS fit as an input to the first step of OLS combination. Additionally, fitted values of KNN regression are not included since KNN regression does not purport creation of fitted values.

3. We carry out an out-of-sample forecast, where inputs for our OLS model are outputs from relevant out-of-sample forecasts from standalone forecasts.

ARIMA on combined fit is basically an ARIMA model, where combined fit (obtained from the OLS combination) is used as a time-series for which ARIMA modelling is done. Diebold (1988) warns that residuals from OLS combination tend to be auto-correlated and this issue should be handled properly since it indicates that there is still some information hidden in the residuals. In our application, we take combined fit as an input and then run ARIMA. For ARIMA model selection, we use the same method as for standalone ARIMA model, that is Hyndman & Khandakar (2008) algorithm (described in Subsection 4.1.2). Once model selection is finalized, we simply create an out-of-sample forecast.

NNAR on combined fit is same as NNAR described in Section 4.2. In this case, combined fit from OLS combination is used as an input for NNAR and output is the out-of-sample forecast.

KNN on combined fit is same as KNN described in Section 4.2. In this case, combined fit from OLS combination is used as an input for KNN and output is the out-of-sample forecast.

4.4 Forecasts evaluation and comparison

Once we have all our forecasts and combined forecasts, we need to assess their performance and compare them in order to determine the most efficient forecasting technique and what volatility proxy to use in each particular case we cover (two training sub-samples and two forecasting horizons).

First of all, since we have 20 stocks, we have 320[7] sets of forecasts, thence, we need to aggregate them in order to have a better picture. Aggregation is done on stocks level. That is, for each stock we have 15 forecasting methods[8], 2 training sub-samples, 2 forecast horizons and their forecasting performance (RMSE, MAE and MAPE). So for each training sub-sample and forecast horizon we

[7] 320=20x4x2x2 — number of stocks x number of volatility proxies x number of training sub-samples x number of forecast horizons.

[8] 10 standalone forecasting methods + 5 forecast combinations.

calculate simple averages of RMSE, MAE and MAPE of forecasting methods across 20 stocks.

$$\text{avgRMSE} = \frac{\sum_i^N \text{RMSE}_i}{N} \tag{4.24}$$

where N is a number of stocks (20). Average RMSE (as well as average MAPE and MAE) of a forecasting method is calculated separately for each forecast horizon, training sub-sample and volatility proxy. For average MAPE and average MAE, similar formula is used[9]. Closer average RMSE, MAE and MAPE to zero indicate a more accurate forecast.

Since we have four volatility proxies, we need to select the most efficient proxy among them. We define 'proxy efficiency' as its level of predictability, that is, we are able to predict (or forecast) this volatility proxy with lowest forecasting errors. Since volatility proxies have different scales[10], we cannot utilize RMSE or MAE, because squared return volatility proxy is likely to have smallest RMSE or MAE due to its scale. Therefore, we need to use a forecast evaluation method that is dependent on relative values, such as MAPE. Thus, determination of the most efficient proxy is based on the average MAPE of the best forecasting method, i.e. we compare lowest average MAPE for each volatility proxy (separately for each training sub-sample and forecast horizon).

Additionally, in order to determine the best forecasting method within one volatility proxy, one training sub-sample and one forecast horizon, we compare average RMSE of a forecasting method and average rank of a forecasting method. Average rank is defined as a simple average of ranks of a forecasting method according to RMSE (from lowest to highest) across 20 stocks[11].

$$\text{avg rank} = \frac{\sum_i^N \text{rank}_i}{N} \tag{4.25}$$

where N is number of stocks (20). Value of avg rank can theoretically be between 1 and 15, since inputs are bounded by 1 and 15. So the best forecasting method will be the one which has value closest to 1.

[9] We also provide standard errors of averages in Appendix A.

[10] Figure 3.5 displays that squared return volatility proxy has much smaller scale than other three volatility proxies.

[11] We also provide standard errors of averages in Appendix A.

Chapter 5

Empirical results

5.1 Empirical Results

This section provides empirical results of our models. The section is divided in two subsections:

1. Example results for one stock.

2. Aggregated results for all stocks we use.

Since in Subsection 5.1.2, we provide aggregated results for 20 stocks, we firstly would like to show what kind of data is being aggregated and what data we get through whole process of model combination in Subsection 5.1.1. Afterwards, in Subsection 5.1.2, we present aggregated results of all 20 stocks.

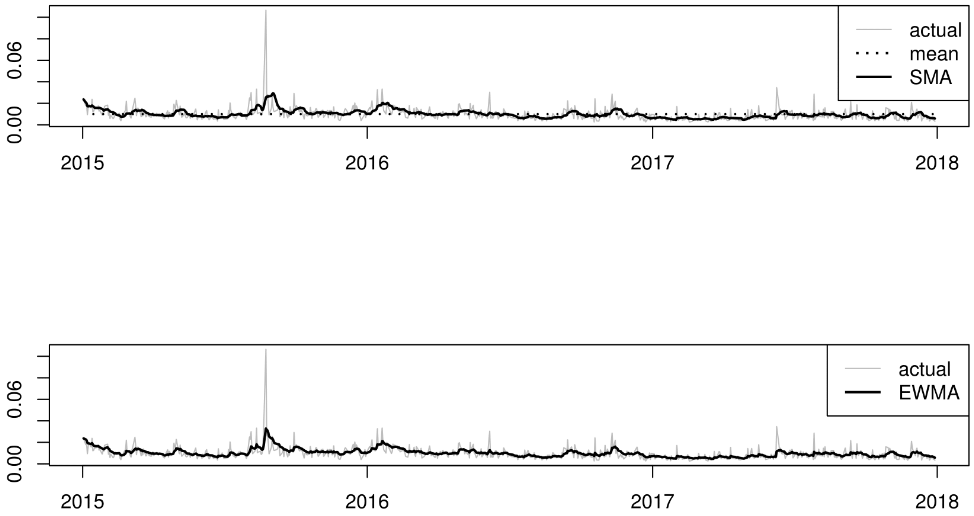
5.1.1 Example results for one stock

This section should serve you as an illustration of what results we obtain for each stock and volatility proxy. Afterwards, these results are aggregated and presented in Subsection 5.1.2. We do not discuss results in this subsection since they are only for one stock, one training sub-sample and one volatility proxy. For illustration purpose, we selected AAPL stock with in-sample data from January 2015 until December 2017 and Garman-Klass estimate as a volatility proxy. On Figure 3.4, our in-sample data is presented, from it, it is seen that there was a spike of Garman-Klass volatility estimate in the second half of 2015. Other than that, Garman-Klass volatility estimates are below 0.04 units. Based on these data, we run all our models.

Since simple models (detailed description in Subsection 4.1.1) do not have any other output rather than fitted values and forecast itself, we do not provide

R output for these 'models' but only plots. Upper plot of Figure 5.1 shows actual Garman-Klass volatility estimate versus its **mean** and its **SMA** fit. Lower plot of Figure 5.1 shows actual Garman-Klass volatility estimate versus its **EWMA** fit. From Figure 5.1, it is well seen how SMA and EWMA are not able to capture the spike in the second half of 2015, however, it is not expected from such simple models.

Figure 5.1: Jan 2015 – Dec 2017 AAPL Garman-Klass volatility estimates versus its mean, SMA and EWMA.



Source: Authors calculations.

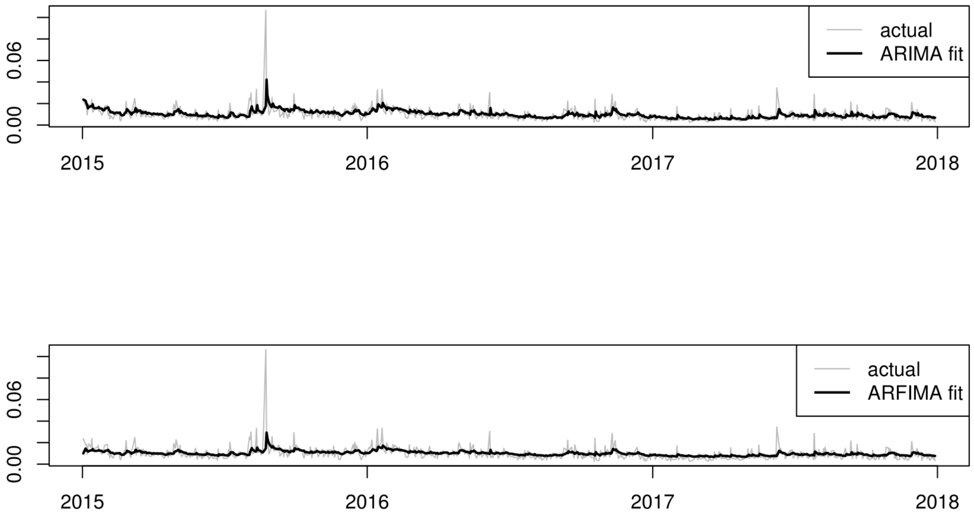
AR(F)IMA models (detailed description in Subsection 4.1.2) outputs are presented in boxes below.

R output of ARIMA model is presented below:

```
ARIMA(1,1,2)


Coefficients:
         ar1      ma1     ma2
      0.5667  -1.2706  0.3055
s.e.  0.1725   0.1865  0.1622


sigma^2 estimated as 2.996e-05:  log likelihood=2857.57
AIC=-5707.14   AICc=-5707.09   BIC=-5688.64
```

From the above output, it is seen that Hyndman-Khandakar algorithm ended up with ARIMA (1,1,2) model. Upper plot of Figure 5.2 shows actual Garman-Klass volatility estimate versus its **ARIMA** fit. Lower plot of Figure 5.2 shows actual Garman-Klass volatility estimate versus its **ARFIMA** fit. From it, we can see that both ARIMA and ARFIMA fits underestimate spikes and ARFIMA fit generally has lower values than ARIMA fit.

Figure 5.2: Jan 2015 – Dec 2017 AAPL Garman-Klass volatility estimates versus ARIMA and ARFIMA fits.



Source: Authors calculations.

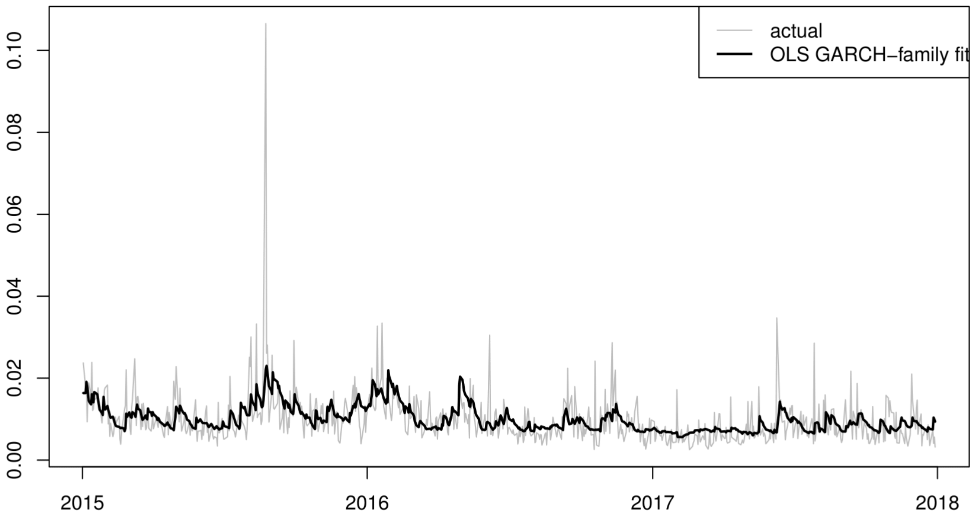
R output of ARFIMA model is presented below:

```
Mean Model : ARFIMA(2,d,2)
Distribution : sstd


Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.009748    0.000630  15.4611 0.000000
ar1     0.000000          NA       NA       NA
ar2    -0.855603    0.037221 -22.9874 0.000000
ma1    -0.035841    0.018004  -1.9907 0.046516
ma2     0.875337    0.020084  43.5844 0.000000
arfima  0.199917    0.035006   5.7110 0.000000
sigma   0.004856    0.000485  10.0094 0.000000
skew    1.977376    0.239497   8.2564 0.000000
shape   3.582542    0.690102   5.1913 0.000000


LogLikelihood : 3180.318
AIC=-8.4035 BIC=-8.3545 HQIC=-8.3846
```

From the above output, we see that all parameters are statistically significant. `arfima` parameter is parameter *d* from ARFIMA model. `skew` parameter represents skewness parameter of the fit. Positive skewness parameter means that a mode of distribution is lower than its mean and this is expected from volatility data, since it is very close to zero and non-negative. `shape` parameter represents kurtosis value. Kurtosis value is greater than 3 (which is a value for a normally distributed variable), which is expected for a financial data since financial data generally have leptokurtic distribution).

Since we run 8 GARCH-family models, we do not provide outputs for each model. Instead, we provide results of the **OLS fit of 8 GARCH-family models**. In the OLS model, the dependent variable is actual Garman-Klass volatility estimate and the independent variables are fitted values of GARCH-family models. In the box below, the output of the OLS model is presented. From the output, it is seen that ALLGARCH, NAGARCH and GARCH have statistically significant estimates. That is, AVGARCH, GJRGARCH, TGARCH, NGARCH and APARCH are not statistically significant, from this, we may conclude that these models are not important in Garman-Klass volatility estimate fit, in other words, these models are not able to capture Garman-Klass volatility well. Additionally, on Figure 5.3 you can see the fit of the OLS model versus actual Garman-Klass volatility estimates, from it, it is seen that the spike of Garman-Klass volatility estimate in the second half of 2015 is underestimated.

Figure 5.3: Jan 2015 – Dec 2017 AAPL Garman-Klass volatility estimates versus OLS GARCH-family fitted values.



Source: Authors calculations.

R output of the OLS model is presented below:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.001950   0.001147  -1.701   0.0894 .
GARCH        0.259295   0.150964   1.718   0.0863 .
AVGARCH     -0.056445   0.240663  -0.235   0.8146
GJRGARCH    -2.386800   1.470084  -1.624   0.1049
TGARCH      -1.274854   1.026483  -1.242   0.2146
NGARCH       0.293616   0.199343   1.473   0.1412
NAGARCH     -0.686261   0.365578  -1.877   0.0609 .
APARCH       3.299995   2.363838   1.396   0.1631
ALLGARCH     1.436856   0.339971   4.226 2.67e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1

Residual standard error: 0.005328 on 746 degrees of freedom
Multiple R-squared:  0.2581,Adjusted R-squared:  0.2501
F-statistic: 32.43 on 8 and 746 DF,  p-value: < 2.2e-16
```

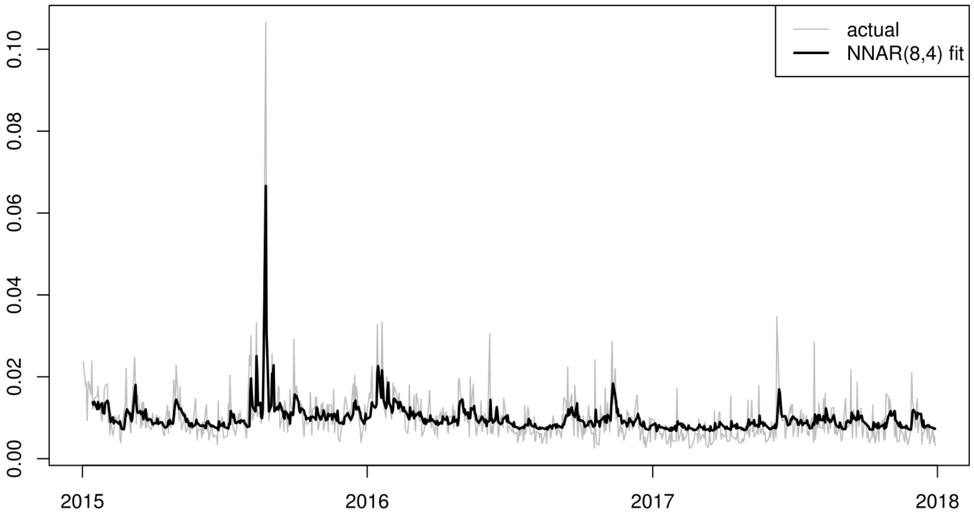
For AAPL Garman-Klass volatility estimate, **NNAR(8,4)** was automatically selected. NNAR(8,4) is a model where 8 lagged values are used as inputs and 4 represents the number of hidden nodes (there is only one hidden layer in NNAR). On Figure 5.4 you can see NNAR(8,4) fit versus actual Garman-Klass volatility estimates, as you can see from it, even the spike in Garman-Klass volatility estimate in the second half of 2015 is well captured.

R output of the NNAR model is presented below:

```
Model:  NNAR(8,4)

Average of 20 networks, each of which is
a 8-4-1 network with 41 weights
options were - linear output units

sigma^2 estimated as 1.758e-05
```

KNN regression output, which is presented below, shows summary of the KNN regression model used for forecasting AAPL Garman-Klass volatility esti-

Figure 5.4: Jan 2015 – Dec 2017 AAPL Garman-Klass volatility estimates versus NNAR(8,4) fitted values.



Source: Authors calculations.

mate. It is seen that what auto-regressive lags are used[1]. Three KNN regression models were created (with k=3, 5 and 7) with median targets combination and MIMO algorithm for multi-step ahead forecasting. Since KNN regression does not purport fitting but only analysis of actuals and out-of-sample forecasting, we do not provide a plot with actuals versus fitted values.

R output of the KNN regression model is presented below:

```
Multiple-Step Ahead Strategy: MIMO
K (number of nearest neighbors): 3 models with 3, 5 and 7
                                   neighbors respectively
Autoregressive lags: 1 2 3 5 6 8
Number of examples: 718
Targets are combined using the median function.
```

Additionally, we fit two MLP models:

1. **MLP model with 1 hidden layer**, automatic lags selection (based on underlying time series frequency), 20 networks to be trained and median operator to be used for final fit and forecast creation. Upper plot in

[1] Note that not 8 lags, but lags up to 8 with lags 4 and 7 being dropped.

Figure 5.5 shows MLP fitted values versus actuals, from it, it seen that the spike in the second half of 2015 is not captured by the model. From the R output presented in the box below, it is seen that MLP used 3 lags (1-3).

2. **MLP model with 2 hidden layers**, selection of lags from a pre-defined pool of lags (we feed 1-125 lags and then MLP keeps only necessary ones), 20 networks to be trained and mode operator to be used for final fit and forecast creation. Lower plot in Figure 5.5 shows MLP fitted values versus actuals[2], from the plot it is seen that the spike in the second half of 2015 is very well captured. From the R output, it is seen that MLP used 19 lags between lags 1-125 and two hidden layers with 3 and 5 nodes in 1st and 2nd hidden layer respectively.

R output of both MLP models are presented below:

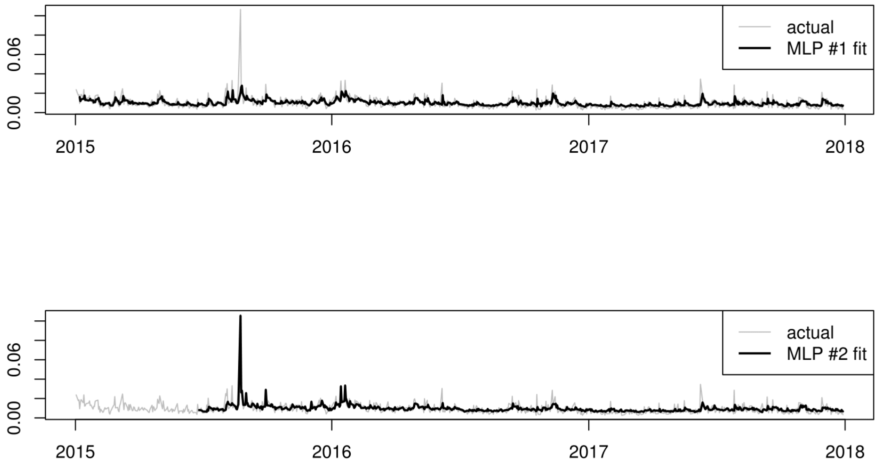
```
MLP fit with 5 hidden nodes and 20 repetitions.
Univariate lags: (1,2,3)
Forecast combined using the median operator.

MLP fit with (3,5) hidden nodes and 20 repetitions.
Univariate lags: (1,2,3,6,8,14,17,25,32,47,53,61,71,82,83,99,
                 102,105,121)
Forecast combined using the mode operator.
```

Once we have all out-of-sample forecasts, we combine them. Figure 5.6, shows boxplots of out-of-sample forecasts from all standalone methods we use versus actual out-of-sample Garman-Klass volatility estimate. That is, we show forecasts ranges versus range of actual data. Mean is higher than the mean of 'actual' boxplot because the mean of in-sample is higher than the mean of out-of-sample. In addition, it is seen that forecast of the MLP model with 1 hidden layer has the largest range and the EWMA forecast is almost flat (since boxplot is flattened). Figure 5.6 well illustrates how various models trained on same sample provide different forecasts and capture almost whole range of actual data.

Results of **simple mean forecast combination** are presented on Figure 5.7, 'other forecasts' are input forecasts used for mean forecast combina-

[2] MLP fit starts from mid-2015 because MLP used values until 121 lag.

Figure 5.5: Jan 2015 – Dec 2017 AAPL Garman-Klass volatility estimates versus MLP fitted values.

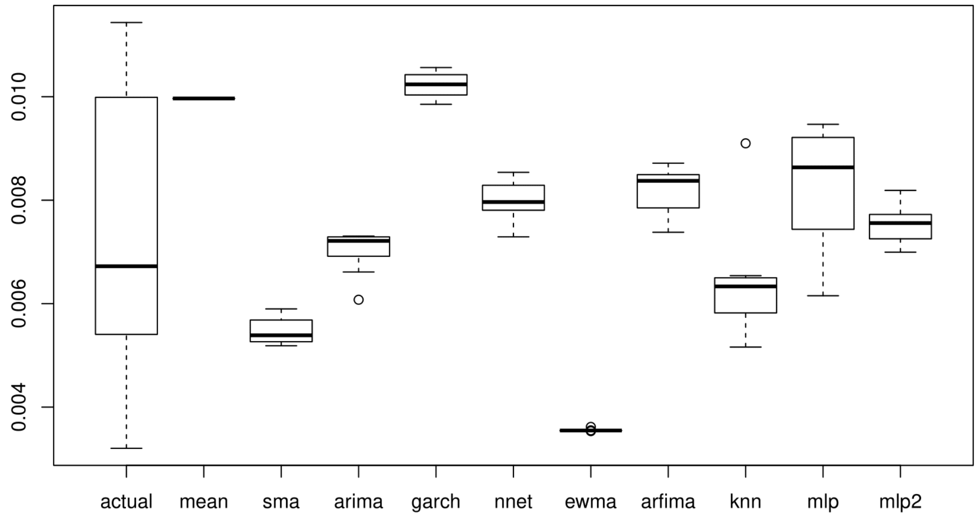


Source: Authors calculations.

tion, that is all standalone forecasts. It is seen that forecast combination is quite flat and unable to capture spikes after January 29.

R output of **OLS combination** is presented in a box below. From it, it is seen that in OLS forecasts combination, GARCH OLS combination (`garch`) and MLP with 2 hidden layers (`mlp2`) are not statistically significant. Additionally, based on high R-squared value (0.964) we may conclude that OLS model combination of our models fits in-sample data very well. We are not interested in values of estimates since our main purpose is to make a fit rather than to explain our data. Figure 5.8 shows OLS combined forecast versus actuals, 'other forecasts' are input forecasts used for OLS forecast combination. It is seen that spikes after January 29 are not captured. It should be noted, that this is due to inability of single forecasts to capture these spikes. Nevertheless, as Diebold (1988) warned, Figure 5.9 shows that residuals of our OLS combined model are auto-correlated. Therefore, ARIMA modelling of OLS fitted values is needed. R output of **ARIMA model on OLS combined fit** is presented in the box below. From the R output, it is seen that Hyndman-Khandakar algorithm ended up with ARIMA (1,1,3) model. Figure 5.10 displays ARIMA (1,1,3) out-of-sample forecast, 'other forecasts' are forecasts from all standalone methods. It is seen that out-of-sample ARIMA forecast converges to a constant and is unable to catch spikes.

Figure 5.6: Boxplots: standalone out-of-sample forecasts versus AAPL Garman-Klass volatility estimates (Jan 1, 2018 – Jan 16, 2018).

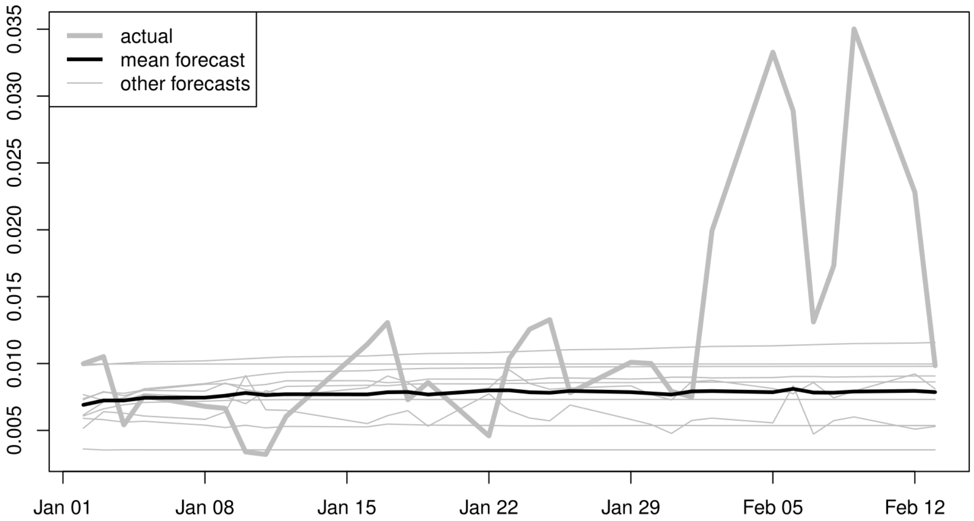


Source: Authors calculations.

R output of the OLS forecast combination:

```
Coefficients:
       Estimate Std. Error t value Pr(>|t|)
mean    0.72926    0.07703   9.468  < 2e-16 ***
sma    -2.18079    0.09166 -23.793  < 2e-16 ***
ewma    4.74416    0.13779  34.431  < 2e-16 ***
arima  -0.78617    0.12777  -6.153 1.36e-09 ***
arfima -1.68714    0.20477  -8.239 1.02e-15 ***
garch  -0.05132    0.05193  -0.988    0.323
nnet    0.61699    0.07657   8.058 3.95e-15 ***
mlp    -0.41403    0.08309  -4.983 8.13e-07 ***
mlp2    0.02915    0.05094   0.572    0.567
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1
Residual standard error: 0.002241 on 625 degrees of freedom
Multiple R-squared:  0.964,Adjusted R-squared:  0.9634
F-statistic:  1858 on 9 and 625 DF,  p-value: < 2.2e-16
```

Figure 5.7: Jan 1, 2018 – Feb 13, 2018 AAPL Garman-Klass volatility estimates versus simple mean forecast combination.



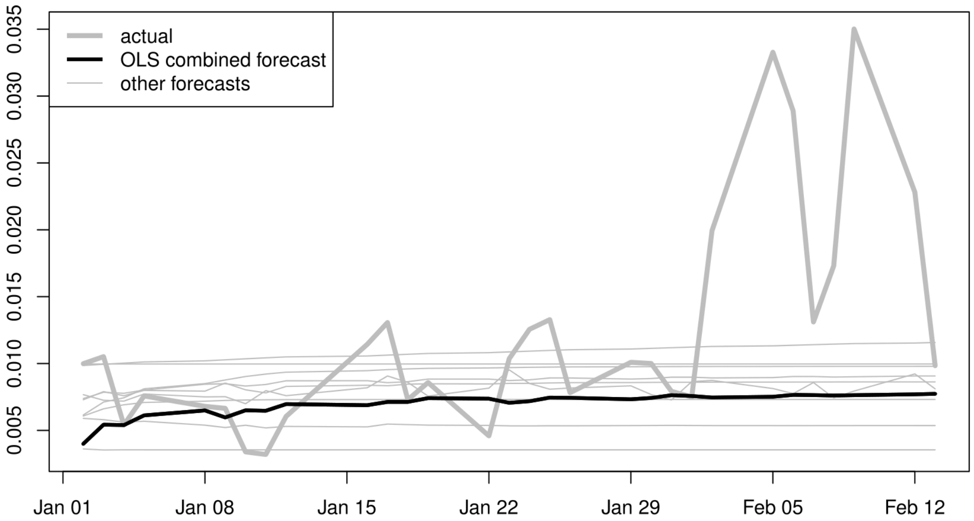
Source: Authors calculations.

R output of the ARIMA model on OLS in-sample fitted values:

```
ARIMA(1,1,3)


Coefficients:
         ar1      ma1     ma2      ma3
      0.8165  -1.5445  0.6073  -0.0566
s.e.  0.0609   0.0710  0.0741   0.0506


sigma^2 estimated as 2.602e-05:  log likelihood=2443.87
AIC=-4877.74   AICc=-4877.64   BIC=-4855.49
```

For OLS combined fit, **NNAR (13,7)** was automatically selected. R output of the NNAR (13,7) model on OLS in-sample combined fit is presented in the box below. Figure 5.11 displays NNAR out-of-sample forecast versus actual data, as seen the forecast range is small and almost flat after Jan 29.

Figure 5.8: Jan 1, 2018 – Feb 13, 2018 AAPL Garman-Klass volatility estimates versus OLS combined forecast.



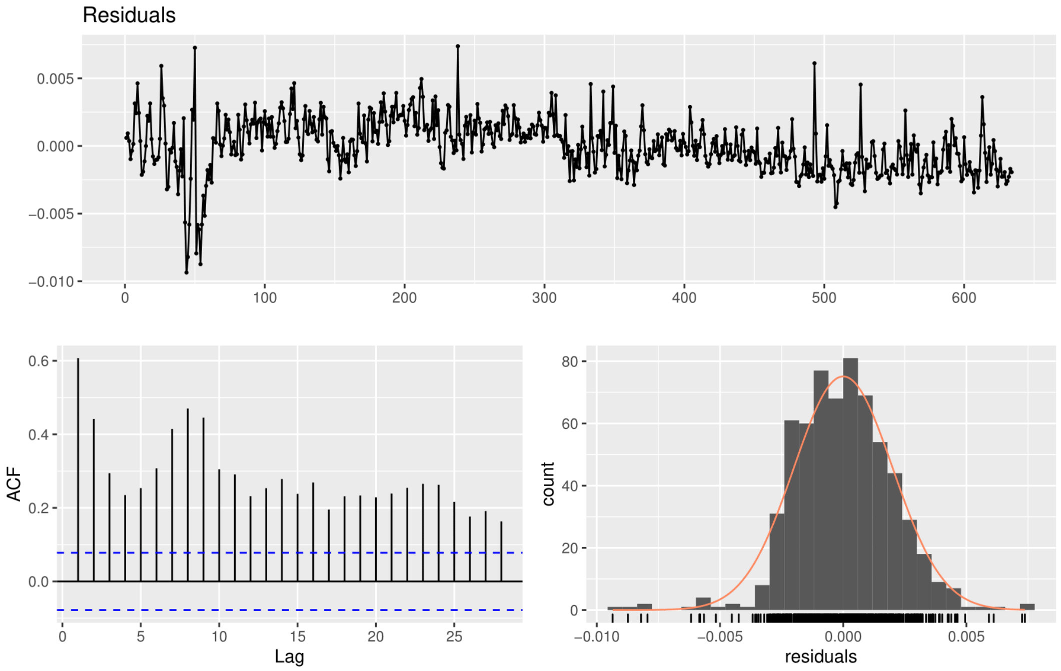
Source: Authors calculations.

R output of the NNAR (13,7) model on OLS in-sample fitted values:

```
Model:  NNAR(13,7)

Average of 20 networks, each of which is
a 13-7-1 network with 106 weights
options were - linear output units

sigma^2 estimated as 6.391e-06
```

The last combination technique is **KNN regression on OLS combined fit**. R output of this KNN regression is presented below. It seen that three KNN regression models were created (with k=3, 5 and 7) with median targets combination and MIMO algorithm for multi-step ahead forecasting. From Figure 5.12, it is seen that the KNN forecast has the biggest range out of all combined forecasts.

Figure 5.9: Analysis of residuals from OLS combination.



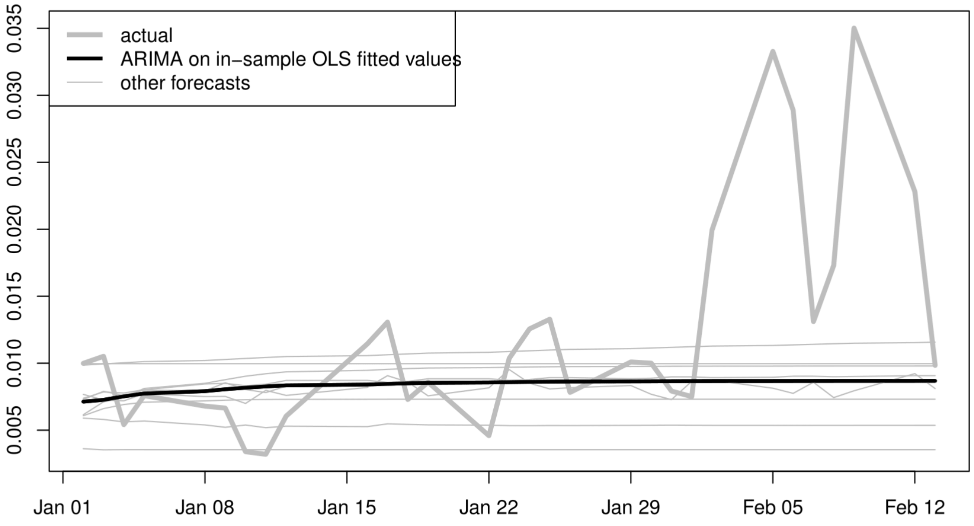
Source: Authors calculations.

R output of the KNN regression model on OLS in-sample fitted values:

```
Multiple-Step Ahead Strategy: MIMO
K (number of nearest neighors): 3 models with 3, 5 and 7
  neighbors repectively
Autoregressive lags: 1 2 3 8 10 11
Number of examples: 594
Targets are combined using the median function.
```

Once we have all standalone forecasts and combined forecasts, we rank them according to RMSE. Table 5.1 presents ranking of out-of-sample forecasts based on first 5 observations (that reflects one business week) according to RMSE. Forecast of NNAR on combined fit has the lowest RMSE. Other forecasts from combination techniques are ranked on 5th (Simple mean combined forecast), 6th (ARIMA on combined fit), 13th (KNN regression on combined fit) and 14th (out-of-sample OLS prediction). From such results, we see that forecast combination technique provides better results than standalone forecasting methods. Nonetheless, if we rank according to MAPE or MAE, MLP with 2 hidden layers provides the best result and NNAR on combined fit provides second best result. If we increase our out-of-sample forecast to two business

Figure 5.10: Jan 1, 2018 – Feb 13, 2018 AAPL Garman-Klass volatility estimates versus ARIMA forecast on OLS combined fit.



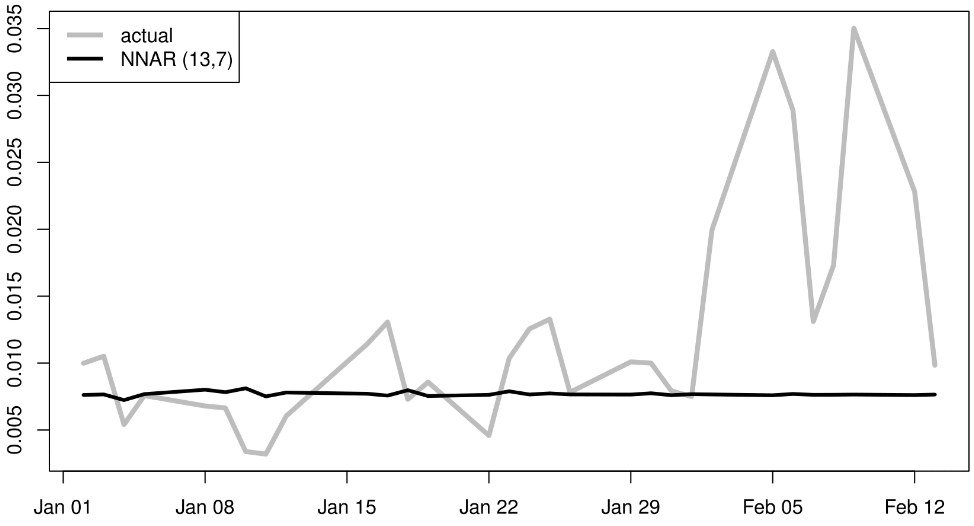
Source: Authors calculations.

weeks (10 observations) results slightly change. Table 5.2 presents the ranking of out-of-sample forecasts for 10 periods according to RMSE. From this ranking, we see that MLP with 2 hidden layers forecasts with the lowest RMSE and MAE. NNAR on combined fit is ranked second accrording to RMSE and MAE. However, according to MAPE, MLP with 2 hidden layers is ranked third, with second being out-of-sample OLS prediction and forecast from SMA being the best one. Therefore, based on the results of only one stock and two out-of-sample horizons, we are unable to conclude that any of combination techniques is superior to a standalone forecasting method. Thus, in Subsection 5.1.2, we aggregate same results for 19 another stocks and based on that information, we try to conclude whether combination techniques provide superior results.

5.1.2 Aggregated results for all stocks

In this subsection, we present aggregated results for all 20 stocks. Since having twenty times Subsection 5.1.1 (with results for each stock) is not informing and it would be very hard to comprehend, we decide to aggregate them, in order to present a clearer picture. Therefore, we aggregate obtained forecasts results

Figure 5.11: Jan 1, 2018 – Feb 13, 2018 AAPL Garman-Klass volatility estimates versus NNAR (13,7) on in-sample OLS combined fit out-of-sample forecast.



Source: Authors calculations.

(such as Table 5.1 and Table 5.2) for all 20 stocks on volatility proxy, forecast horizon and training sub-sample levels. That is, we have aggregated results for 4 volatility proxies, 2 forecast horizons and 2 training sub-samples, which yields 16 (=4x2x2) aggregated results, which is also a lot and hard to comprehend, so we decided to summarize aggregated results in Chapter 6 in order to present our results in a clearer way. First we present results for medium training sub-sample[3], then for short training sub-sample[4], we compare them and summarize in Chapter 6.

Aggregated results for medium training sub-sample

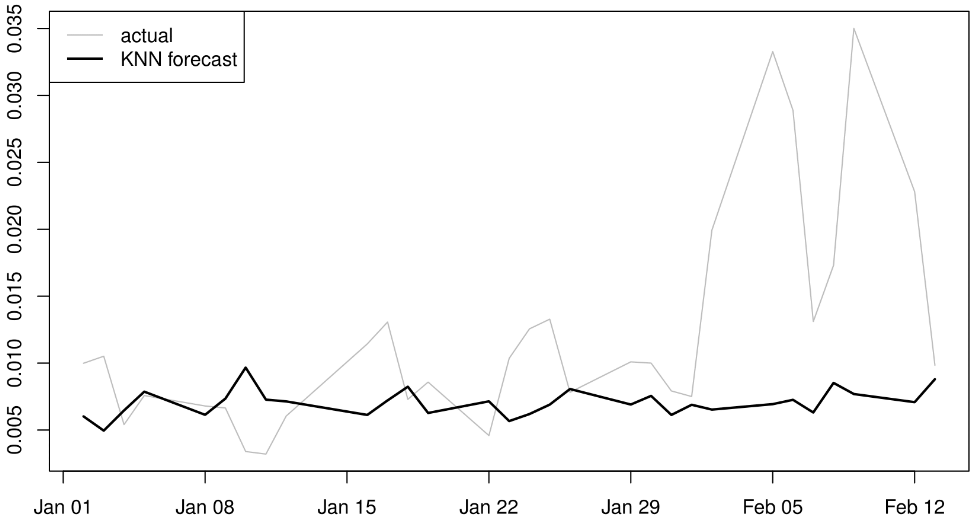
Table 5.3 presents aggregated forecasts results for squared return and absolute return volatility proxies. Table 5.4 presents aggregated results for Parkinson and Garman-Klass volatility proxies. Aggregation was done by simple averaging of RMSE, MAE and MAPE over 20 forecasts[5] on medium training sub-

[3] Training data for our models from Jan 2015 until Dec 2017.

[4] Training data for our models from Jan 2017 until Dec 2017.

[5] That is, for each stock.

Figure 5.12: Jan 1, 2018 – Feb 13, 2018 AAPL Garman-Klass volatility estimates versus KNN regression on in-sample OLS combined fit out-of-sample forecast.



Source: Authors calculations.

sample and forecast horizon set to 5 periods ahead. Efficiency[6] of volatility proxy is compared by lowest average MAPE since comparison by averages of RMSE or MAE may provide misleading results due to different scales of proxies (Figure 3.5 clearly shows that squared return volatility proxy has much smaller scale than other three volatility proxies).

Table 5.5 displays ranking according to average ranks across 20 stocks. Average rank in Table 5.5 is calculated as average of ranks according to RMSE, for instance, ranking from Table 5.1 is one of 20 inputs for obtaining average rank for every method.

As seen from average MAPE columns in Table 5.3 and Table 5.4, squared returns and absolute returns volatility proxies have much higher lowest average MAPE than Parkinson and Garman-Klass volatility proxies. Lowest average MAPE of squared returns and absolute returns volatility proxies are 7895.12 and 296.89 respectively versus 32.66 and 28.32 (lowest average MAPE of Parkinson and Garman-Klass volatility proxies respectively). This means that our best forecasts (according to average MAPE) for squared returns volatil-

[6] We measure volatility proxy efficiency based on its forecast predictability as described in Section 4.4

Table 5.1: Out-of-sample forecasts ranking based on first 5 observations (according to RMSE).

rank	model	RMSE	MAE	MAPE
1	NNAR on combined fit	0.001929	0.001675	20.80592
2	MLP with 2 hidden layers	0.001985	0.001643	19.9166
3	NNAR	0.002018	0.00181	22.90764
4	ARFIMA	0.002119	0.001948	25.09004
5	Simple mean combined forecast	0.002192	0.001797	21.4759
6	ARIMA on combined fit	0.002214	0.001901	23.45675
7	ARIMA	0.002577	0.00204	23.27697
8	MLP with 1 hidden layer	0.002591	0.002301	28.08384
9	Mean	0.002715	0.002138	33.59004
10	GARCH-family	0.002822	0.002255	35.19752
11	KNN regression	0.002975	0.002461	27.58988
12	SMA	0.002985	0.002463	27.06964
13	KNN regression on combined fit	0.003107	0.002306	25.12427
14	Out-of-sample OLS prediction	0.003578	0.00257	26.44426
15	EWMA	0.00489	0.004497	53.09573

ity proxy predicts future points with average mean absolute percentage error of 7895.12 and this is for 5 points ahead, which is an unacceptably high error[7]. In this particular sample, according to our definition of volatility proxy efficiency, it is seen that Garman-Klass volatility proxy is the most efficient.

As it is seen from Table 5.4, for Garman-Klass volatility estimate, a forecast from KNN regression on OLS combined fit is the best one having 10% lower average RMSE than the second best that is a forecast from simple mean combined forecast. Additionally, from Table 5.5, it is seen that KNN regression on combined fit, on average, was ranked higher than other forecasting methods. In fact, top three forecasting methods according to average rank are from forecast combinations: (1) KNN regression on combined fit, (2) ARIMA on combined fit and (3) simple mean combined forecast.

Additionally, from Table 5.4, it is seen that rankings may change based on different forecasting assessment measure. For example, for Parkinson volatility proxy: KNN regression on combined fit is ranked first according to average RMSE and MAE, however, according to average MAPE, it is ranked second, with first being SMA.

When we increase forecast horizon from 5 to 10, results, as expected, change. Table 5.6 presents aggregated forecasts results for squared return and absolute

[7] In order to provide some comprehension, imagine temperature forecast for tomorrow is 10 degrees Celsius and actual temperature is 30 degrees, so here MAPE = 300.

Table 5.2: Out-of-sample forecasts ranking based on first 10 observations (according to RMSE).

rank	model	RMSE	MAE	MAPE
1	MLP with 2 hidden layers	0.002629	0.002225	42.16231
2	NNAR on combined fit	0.00279	0.002407	45.70448
3	Simple mean combined forecast	0.002842	0.002417	44.9977
4	NNAR	0.002846	0.002548	48.70887
5	ARIMA	0.002897	0.002414	42.39419
6	ARIMA on combined fit	0.002989	0.002607	50.16016
7	ARFIMA	0.003031	0.002711	52.91605
8	SMA	0.003052	0.002464	34.37766
9	Out-of-sample OLS prediction	0.003261	0.002537	39.08699
10	MLP with 1 hidden layer	0.003474	0.003054	59.77609
11	KNN regression	0.003512	0.002799	47.35274
12	KNN regression on combined fit	0.003661	0.002897	51.23498
13	Mean	0.003928	0.003273	70.0181
14	GARCH-family	0.004173	0.003441	74.22352
15	EWMA	0.004447	0.003647	43.77129

return volatility proxies. Table 5.7 presents aggregated results for Parkinson and Garman-Klass volatility proxies. It is naturally expected that results will deteriorate (or at least change) with increasing forecast horizon, simply because you forecast for a further future points having same information set. Nonetheless, volatility proxy efficiency ranking order remains same: Garman-Klass volatility proxy has the lowest average MAPE (33.17), then Parkinson volatility proxy (33.81), then absolute returns and squared returns volatility proxies (298.56 and 13659.94, respectively). Additionally, though ranking of forecasts according to average RMSE change, for Garman-Klass volatility proxy, KNN regression on combined fit still provided forecasts with the lowest average RMSE (as with forecast horizon = 5). Nevertheless, for remaining three volatility proxies, models providing best forecasts according to the lowest average RMSE changed. But, if we look at average rank (Table 5.8), for Garman-Klass volatility proxy, we see that ARIMA forecast has, on average, higher rank than others (according to RMSE) and forecasts from KNN regression on combined fit are, on average, are ranked sixth. This may seem counter-intuitive, but this is caused by imperfections of average rank, which is a simple average of ranks and simple average can be strongly affected by outliers (in this case, it can be caused by fact that for some stock, KNN regression on combined fit was ranked worst and that's why, average rank is lower, also, we need to take into account standard errors (Table A.6)).

Nonetheless, for medium training sub-sample we see that Garman-Klass volatility proxy is the most efficient and forecasts from KNN regression on combined fit provide lowest average RMSE and MAE for both forecast horizons[8]. Next, we present similar set of results for short training sub-sample.

[8] Standard errors for construction of confidence intervals are provided in Appendix A.

Table 5.3: Forecasts aggregation on medium training sub-sample and forecast horizon = 5 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy				**Absolute return volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	14	0.0013	0.0013	28472.17	12	0.0100	0.0091	475.30
ARIMA	6	0.0005	0.0005	20332.38	5	0.0091	0.0080	378.64
ARIMA on combined fit	5	0.0005	0.0004	29796.06	9	0.0098	0.0085	405.64
EWMA	2	0.0003	0.0002	12033.10	8	0.0097	0.0082	306.15
GARCH-Family	15	0.0069	0.0069	1585571.29	6	0.0092	0.0077	388.71
KNN regression	**1**	0.0002	0.0002	7895.12	10	0.0098	0.0083	422.14
KNN regression on combined fit	4	0.0004	0.0003	16991.07	2	0.0088	0.0072	296.89
Mean	8	0.0006	0.0006	34885.30	11	0.0100	0.0090	476.23
MLP w/ 1 hidden layer	7	0.0006	0.0006	34553.16	4	0.0090	0.0081	443.48
MLP w/ 2 hidden layers	9	0.0007	0.0006	28133.29	14	0.0106	0.0094	319.04
NNAR	11	0.0010	0.0009	31370.85	**1**	0.0087	0.0075	408.65
NNAR on combined fit	10	0.0008	0.0006	25458.05	13	0.0102	0.0087	480.03
Out-of-sample OLS prediction	12	0.0011	0.0009	32327.23	15	0.0144	0.0130	605.01
Simple mean combined forecast	13	0.0012	0.0011	179398.30	3	0.0089	0.0077	390.20
SMA	3	0.0003	0.0003	11744.48	7	0.0092	0.0078	298.31

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.1.

Table 5.4: Forecasts aggregation on medium training sub-sample and forecast horizon = 5 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy				**Garman-Klass volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	10	0.0051	0.0043	38.63	3	0.0044	0.0038	33.59
ARIMA	5	0.0049	0.0039	33.27	4	0.0045	0.0037	30.80
ARIMA on combined fit	9	0.0051	0.0042	36.67	8	0.0048	0.0041	34.17
EWMA	14	0.0060	0.0051	40.89	13	0.0060	0.0052	39.62
GARCH-Family	11	0.0051	0.0042	35.60	9	0.0050	0.0042	33.94
KNN regression	8	0.0050	0.0041	33.50	10	0.0051	0.0042	35.67
KNN regressionon combined fit	**1**	0.0046	0.0038	32.95	**1**	0.0040	0.0033	28.32
Mean	13	0.0057	0.0051	49.37	11	0.0054	0.0048	45.57
MLP w/ 1 hidden layer	4	0.0048	0.0039	35.37	7	0.0048	0.0040	35.62
MLP w/ 2 hidden layers	3	0.0047	0.0038	38.81	14	0.0065	0.0050	40.82
NNAR	6	0.0050	0.0041	37.73	5	0.0046	0.0038	33.12
NNAR on combined fit	12	0.0055	0.0045	42.51	12	0.0056	0.0049	45.40
Out-of-sample OLS prediction	15	0.0106	0.0097	78.03	15	0.0084	0.0074	57.46
Simple mean combined forecast	2	0.0047	0.0038	33.68	2	0.0044	0.0037	31.18
SMA	7	0.0050	0.0041	32.66	6	0.0047	0.0040	31.11

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.2.

Table 5.5: Average ranks of forecasts on medium training sub-sample and forecast horizon = 5.

	Squared returns volatility proxy		**Absolute returns volatility proxy**		**Parkinson volatility proxy**		**Garman-Klass volatility proxy**	
model	rank	avg rank	rank	avg rank	rank	avg rank	rank	avg rank
ARFIMA	11	8.1	2	6.0	3	6.3	4	6.4
ARIMA	**1**	6.0	4	7.1	5	7.0	5	6.7
ARIMA on combined fit	4	6.9	10	8.1	4	6.5	2	6.1
EWMA	13	9.0	3	7.0	14	10.0	14	10.3
GARCH-Family	7	7.4	15	12.3	6	7.3	7	7.0
KNN regression	12	8.3	**1**	5.3	10	7.9	13	10.1
KNN regression on combined fit	8	7.7	5	7.1	**1**	5.4	**1**	6.0
Mean	10	8.0	13	8.7	12	8.4	11	8.8
MLP with 1 hidden layer	6	7.2	12	8.5	7	7.7	10	8.5
MLP with 2 hidden layers	5	7.2	6	7.5	8	7.7	8	7.2
NNAR	3	6.5	9	7.9	9	7.8	6	6.8
NNAR on combined fit	14	9.0	8	7.6	13	8.6	9	8.3
Out-of-sample OLS prediction	15	11.0	11	8.2	15	12.8	15	12.2
Simple mean combined forecast	2	6.4	14	11.0	2	5.6	3	6.3
SMA	9	7.9	7	7.6	11	8.2	12	8.9

Note: Ranking according to avg rank (lowest avg rank – 1, highest avg rank – 15). Average rank is calculated as a simple average of ranks (according to RMSE) across 20 stocks. Standard errors of averages are in Table A.3

Table 5.6: Forecasts aggregation on medium training sub-sample and forecast horizon = 10 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy				**Absolute return volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	14	0.0015	0.0013	35196.52	12	0.0111	0.0091	498.35
ARIMA	6	0.0006	0.0005	25368.42	**1**	0.0099	0.0080	396.17
ARIMA on combined fit	5	0.0006	0.0005	31784.08	6	0.0104	0.0085	424.74
EWMA	2	0.0005	0.0003	13659.94	13	0.0112	0.0084	323.01
GARCH-Family	15	0.0072	0.0071	980923.67	8	0.0107	0.0083	439.44
KNN regression	3	0.0005	0.0003	17523.75	10	0.0110	0.0084	370.22
KNN regression on combined fit	4	0.0006	0.0004	22002.24	2	0.0100	0.0076	337.61
Mean	7	0.0007	0.0006	41324.38	11	0.0111	0.0091	501.53
MLP w/ 1 hidden layer	8	0.0007	0.0006	40896.06	7	0.0106	0.0086	500.71
MLP w/ 2 hidden layers	9	0.0008	0.0006	32964.16	15	0.0148	0.0111	493.98
NNAR	11	0.0010	0.0009	35863.41	3	0.0101	0.0079	418.94
NNAR on combined fit	10	0.0008	0.0007	31123.94	9	0.0109	0.0086	449.83
Out-of-sample OLS prediction	12	0.0011	0.0009	32093.58	14	0.0142	0.0122	535.11
Simple mean combined forecast	13	0.0013	0.0012	123734.84	4	0.0101	0.0079	412.59
SMA	**1**	0.0004	0.0003	15576.51	5	0.0103	0.0078	298.56

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.4.

Table 5.7: Forecasts aggregation on medium training sub-sample and forecast horizon = 10 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy				**Garman-Klass volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	9	0.0058	0.0048	43.56	4	0.0053	0.0044	38.90
ARIMA	2	0.0052	0.0041	36.35	2	0.0049	0.0039	33.91
ARIMA on combined fit	4	0.0055	0.0044	40.47	5	0.0053	0.0043	38.10
EWMA	14	0.0068	0.0055	42.60	14	0.0069	0.0057	43.31
GARCH-Family	10	0.0058	0.0046	40.52	9	0.0056	0.0045	38.20
KNN regression	8	0.0057	0.0045	37.18	6	0.0054	0.0043	36.15
KNN regression on combined fit	5	0.0056	0.0044	39.33	**1**	0.0049	0.0039	33.45
Mean	13	0.0066	0.0057	54.51	12	0.0063	0.0055	51.36
MLP w/ 1 hidden layer	11	0.0059	0.0048	45.01	10	0.0057	0.0048	42.69
MLP w/ 2 hidden layers	**1**	0.0050	0.0040	40.55	13	0.0064	0.0048	39.80
NNAR	6	0.0057	0.0047	41.78	7	0.0055	0.0045	38.51
NNAR on combined fit	12	0.0060	0.0050	47.73	11	0.0063	0.0052	48.94
Out-of-sample OLS prediction	15	0.0106	0.0094	78.03	15	0.0090	0.0077	64.34
Simple mean combined forecast	3	0.0054	0.0043	37.86	3	0.0051	0.0042	35.43
SMA	7	0.0057	0.0044	33.81	8	0.0055	0.0044	33.17

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.5.

Table 5.8: Average ranks of forecasts on medium training sub-sample and forecast horizon = 10.

	Squared returns volatility proxy		**Absolute returns volatility proxy**		**Parkinson volatility proxy**		**Garman-Klass volatility proxy**	
model	rank	avg rank	rank	avg rank	rank	avg rank	rank	avg rank
ARFIMA	**1**	5.8	8	7.7	2	5.8	3	6.2
ARIMA	2	6.6	**1**	4.6	**1**	5.3	**1**	5.8
ARIMA on combined fit	6	7.4	4	6.8	4	6.2	2	6.0
EWMA	8	8.0	13	9.1	14	10.4	14	11.4
GARCH-Family	15	10.6	9	7.7	5	6.8	5	6.7
KNN regression	3	6.6	14	10.1	13	8.8	9	8.0
KNN regression on combined fit	10	8.3	6	7.1	7	7.2	6	7.1
Mean	13	9.2	10	8.1	12	8.6	13	9.2
MLP with 1 hidden layer	11	8.7	5	7.0	8	7.5	10	8.2
MLP with 2 hidden layers	7	7.9	12	9.1	6	6.9	7	7.3
NNAR	9	8.2	3	6.6	9	7.8	8	7.5
NNAR on combined fit	4	6.7	11	8.7	10	8.1	11	8.3
Out-of-sample OLS prediction	12	9.0	15	11.4	15	13.2	15	13.2
Simple mean combined forecas	14	10.2	2	6.0	3	6.2	4	6.2
SMA	5	6.9	7	7.5	11	8.4	12	8.5

Note: Ranking according to avg rank (lowest avg rank – 1, highest avg rank – 15). Average rank is calculated as a simple average of ranks (according to RMSE) across 20 stocks. Standard errors of averages are in Table A.6

Aggregated results for short training sub-sample

Table 5.9 and Table 5.10 present aggregated forecast results for models trained on short sub-sample and forecast horizon set to 5 periods ahead. Aggregation is done in same way as in results for medium training sub-sample.

The most efficient volatility proxy (according to lowest average MAPE) is Garman-Klass volatility proxy with the lowest average MAPE equal to 27.65, then Parkinson, absolute returns and squared returns volatility proxies (lowest average MAPE are 32.79, 300 and 10947 respectively). Ordering of volatility proxy efficiency remains same as in models with medium training sub-sample, however models that provide lowest average MAPE change, this will be discussed in Chapter 6.

From Table 5.10, for Garman-Klass volatility proxy, it is seen that according to RMSE, the most accurate forecast is provided by simple mean, i.e. a simple mean[9] has lower average RMSE than all other models. However, if we take into account standard errors of averages (Table A.8), we can calculate that confidence intervals of average RMSE for forecasts provided by simple mean and by MLP with 2 hidden layers overlap. Nonetheless, taking into account just point estimate of average RMSE, mean provides the most accurate forecast, though we hardly can call it a forecast. Second and third most accurate forecasts according to average RMSE are both MLP models (with 2 and 1 hidden layer respectively). From set of combination techniques, the most accurate forecast (according to average RMSE) is provided by ARIMA on combined fit with seventh most accurate forecast.

In addition, if we look at average ranks in Table 5.11, we can see that on average, ’forecasts’ from simple mean are ranked higher than others for all four volatility proxies.

Nevertheless, if we measure forecast accuracy by average MAPE, top three reshuffles: with MLP w/1 hidden layer being the most accurate and MLP w/2 hidden layers and simple mean being second and third most accurate forecasts respectively.

When we increase our forecast horizon from 5 to 10 periods ahead, results do not change structurally (Table 5.12, Table 5.13 and Table 5.14). In terms of volatility efficiency, order of ranking is same: Garman-Klass (with lowest average MAPE of 29.04), Parkinson (32.79), absolute return (368) and

[9] Note that this is not a simple mean forecast combination but a usual historical in-sample mean.

squared return (14535). For Garman-Klass volatility proxy, top 5 most accurate forecasts according to average RMSE are provided by same methods as when forecast horizon set to 5 with only minor reshuffling. Again, in terms of average RMSE, forecasts from simple mean provide lowest average RMSE, but same as with forecast horizon set to 5, if we take into account standard errors (Table A.11), we can calculate that confidence intervals of average RMSE for forecasts provided by simple mean and by MLP with 1 hidden layer overlap significantly. In any case, if we take point estimates, simple mean again provides the most accurate forecast according to RMSE.

On the other hand, if we rank according to another forecast measure, e.g. average MAE, then MLP with 2 hidden layers provides most accurate forecasts. And if we rank according to average MAPE, then MLP with 1 hidden layer provides most accurate forecasts.

Additionally, Table 5.14, shows that top three (according to average rank) remain same as when forecast horizon set to 5.

Table 5.9: Forecasts aggregation on short training sub-sample and forecast horizon = 5 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy				**Absolute return volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	7	0.00064	0.000409	10947	4	0.0113	0.0091	374.45
ARIMA	5	0.00058	0.000463	22422	5	0.0114	0.0093	383.13
ARIMA on combined fit	11	0.00065	0.00050	21728	7	0.0119	0.0095	358.12
EWMA	9	0.00064	0.00040	11090	11	0.0127	0.0096	319.13
GARCH-Family	15	0.01194	0.01177	1252624	12	0.0129	0.0106	430.62
KNN regression	10	0.00065	0.00041	14883	14	0.0133	0.0102	325.92
KNN regression on combined fit	6	0.00063	0.000428	16148	9	0.0122	0.0097	300.27
Mean	3	0.00058	0.000442	21851	2	0.0112	0.0090	372.69
MLP w/ 1 hidden layer	**1**	0.00057	0.000438	21067	6	0.0119	0.0095	354.19
MLP w/ 2 hidden layers	2	0.00058	0.000438	21884	**1**	0.0112	0.0089	361.23
NNAR	4	0.00058	0.000434	21172	3	0.0113	0.0090	372.39
NNAR on combined fit	12	0.00083	0.000663	32353	13	0.0130	0.0106	357.13
Out-of-sample OLS prediction	13	0.00084	0.000714	66340	15	0.0180	0.0153	762.23
Simple mean combined forecast	14	0.00130	0.001229	144022	10	0.0124	0.0098	369.15
SMA	8	0.00064	0.00044	24053	8	0.0121	0.0095	350.66

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.7.

Table 5.10: Forecasts aggregation on short training sub-sample and forecast horizon = 5 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy				**Garman-Klass volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	3	0.0062	0.0050	32.44	5	0.0056	0.0047	29.47
ARIMA	7	0.0066	0.0055	37.04	7	0.0059	0.0049	32.68
ARIMA on combined fit	5	0.0065	0.0051	32.19	6	0.0058	0.0049	34.08
EWMA	13	0.0076	0.0064	40.17	14	0.0071	0.0062	39.47
GARCH-Family	14	0.0078	0.0064	43.45	13	0.0069	0.0058	40.31
KNN regression	10	0.0071	0.0060	38.50	11	0.0065	0.0054	33.81
KNN regression on combined fit	12	0.0072	0.0060	38.00	10	0.0062	0.0052	33.68
Mean	1	0.0059	0.0048	32.00	**1**	0.0051	0.0042	28.33
MLP w/ 1 hidden layer	**4**	0.0062	0.0050	31.26	3	0.0053	0.0044	27.65
MLP w/ 2 hidden layers	2	0.0060	0.0048	31.53	2	0.0052	0.0043	27.93
NNAR	6	0.0065	0.0055	34.54	4	0.0055	0.0046	28.82
NNAR on combined fit	9	0.0070	0.0058	37.98	8	0.0060	0.0050	33.71
Out-of-sample OLS prediction	15	0.0106	0.0093	69.63	15	0.0118	0.0109	75.69
Simple mean combined forecast	8	0.0069	0.0057	33.55	9	0.0060	0.0049	29.40
SMA	11	0.0072	0.0062	37.97	12	0.0065	0.0056	34.46

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.8.

Table 5.11: Average ranks of forecasts on short training sub-sample and forecast horizon = 5.

	Squared return volatility proxy		**Absolute return volatility proxy**		**Parkinson volatility proxy**		**Garman-Klass volatility proxy**	
model	rank	avg rank	rank	avg rank	rank	avg rank	rank	avg rank
ARFIMA	4	5.7	5	6.3	4	5.7	4	6.5
ARIMA	6	6.8	2	5.8	7	6.7	7	7.1
ARIMA on combined fit	12	9.0	8	7.8	6	6.4	6	7.0
EWMA	9	7.6	13	10.2	14	10.7	14	11.7
GARCH-Family	15	14.5	12	9.1	11	9.6	11	8.9
KNN regression	11	8.5	14	10.4	13	10.2	12	9.7
KNN regression on combined fit	8	7.4	11	8.8	10	9.4	10	8.6
Mean	**1**	5.5	**1**	5.1	**1**	4.3	**1**	4.3
MLP w/ 1 hidden layer	2	5.5	4	6.3	3	5.6	2	5.4
MLP w/ 2 hidden layers	5	6.1	6	6.4	2	5.4	3	5.4
NNAR	3	5.7	3	6.2	5	6.3	5	6.5
NNAR on combined fit	10	8.3	9	7.9	9	8.3	9	7.4
Out-of-sample OLS prediction	13	9.4	15	13.2	15	12.8	15	12.9
Simple mean combined forecast	14	12.2	10	8.6	8	6.8	8	7.4
SMA	7	7.3	7	7.7	12	10.0	13	10.2

Note: Ranking according to avg rank (lowest avg rank – 1, highest avg rank – 15). Average rank is calculated as a simple average of ranks (according to RMSE) across 20 stocks. Standard errors of averages are in Table A.9.

Table 5.12: Forecasts aggregation on short training sub-sample and forecast horizon = 10 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy				**Absolute return volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	7	0.00059	0.00036	14693	2	0.0110	0.0088	464.40
ARIMA	5	0.00056	0.00044	33669	4	0.0113	0.0090	476.46
ARIMA on combined fit	11	0.00062	0.00047	40298	7	0.0117	0.0091	447.28
EWMA	6	0.00059	0.00035	14535	10	0.0122	0.0090	367.92
GARCH-Family	15	0.01392	0.01356	1362638	14	0.0140	0.0118	528.95
KNN regression	8	0.00060	0.00035	20587	12	0.0124	0.0094	382.04
KNN regression on combined fit	10	0.00062	0.00040	21698	11	0.0124	0.0094	418.38
Mean	4	0.00054	0.00042	32844	**1**	0.0110	0.0087	460.13
MLP w/ 1 hidden layer	**1**	0.00054	0.00041	33069	6	0.0116	0.0092	455.41
MLP w/ 2 hidden layers	2	0.00054	0.00041	38909	5	0.0115	0.0090	461.03
NNAR	3	0.00054	0.00041	30850	3	0.0110	0.0087	452.95
NNAR on combined fit	12	0.00077	0.00056	45543	13	0.0125	0.0100	458.02
Out-of-sample OLS prediction	13	0.00089	0.00077	66719	15	0.0172	0.0146	929.25
Simple mean combined forecast	14	0.00151	0.00144	160778	8	0.0118	0.0093	428.03
SMA	9	0.00061	0.00040	25800	9	0.0120	0.0091	379.24

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.10.

Table 5.13: Forecasts aggregation on short training sub-sample and forecast horizon = 10 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy				**Garman-Klass volatility proxy**			
model	rank	avg RMSE	avg MAE	avg MAPE	rank	avg RMSE	avg MAE	avg MAPE
ARFIMA	3	0.0059	0.0047	34.16	5	0.0057	0.0046	31.85
ARIMA	7	0.0063	0.0052	39.01	8	0.0058	0.0049	35.23
ARIMA on combined fit	6	0.0061	0.0048	34.96	7	0.0058	0.0048	35.83
EWMA	13	0.0074	0.0062	41.75	13	0.0073	0.0062	41.03
GARCH-Family	14	0.0091	0.0076	61.21	14	0.0086	0.0073	56.90
KNN regression	11	0.0068	0.0055	39.21	10	0.0062	0.0050	34.15
KNN regression on combined fit	9	0.0066	0.0053	37.57	11	0.0062	0.0052	34.90
Mean	**1**	0.0056	0.0045	33.79	**1**	0.0051	0.0041	30.30
MLP w/ 1 hidden layer	4	0.0059	0.0046	32.79	2	0.0052	0.0041	29.04
MLP w/ 2 hidden layers	2	0.0057	0.0044	33.07	3	0.0052	0.0041	29.91
NNAR	5	0.0060	0.0048	34.59	4	0.0054	0.0044	30.40
NNAR on combined fit	10	0.0067	0.0055	40.94	6	0.0058	0.0047	34.42
Out-of-sample OLS prediction	15	0.0107	0.0096	77.76	15	0.0120	0.0111	81.48
Simple mean combined forecast	8	0.0065	0.0053	36.69	9	0.0061	0.0050	33.17
SMA	12	0.0071	0.0059	38.70	12	0.0066	0.0056	36.07

Note: Ranking according to RMSE (lowest RMSE – 1, highest RMSE – 15). Averaging is done across 20 stocks. Standard errors of averages are in Table A.11.

Table 5.14: Average ranks of forecasts on short training sub-sample and forecast horizon = 10.

	Squared return volatility proxy		**Absolute return volatility proxy**		**Parkinson volatility proxy**		**Garman-Klass volatility proxy**	
model	rank	avg rank	rank	avg rank	rank	avg rank	rank	avg rank
ARFIMA	5	6.0	4	5.8	3	5.2	6	6.4
ARIMA	6	6.8	3	5.6	7	6.8	7	6.7
ARIMA on combined fit	8	7.5	9	8.4	6	6.6	4	5.9
EWMA	10	7.7	13	10.0	13	10.5	14	11.2
GARCH-Family	15	14.4	12	9.3	11	9.8	12	9.9
KNN regression	12	9.2	14	10.1	14	10.6	11	9.7
KNN regression on combined fit	9	7.5	11	9.3	10	9.1	10	9.3
Mean	3	5.4	**1**	5.3	**1**	4.0	**1**	13.0
MLP w/ 1 hidden layer	2	5.3	5	6.0	4	5.2	2	4.2
MLP w/ 2 hidden layers	4	5.5	7	6.9	2	5.1	3	7.5
NNAR	**1**	4.8	2	5.4	5	6.2	5	4.7
NNAR on combined fit	11	9.0	10	8.8	9	8.9	9	5.6
Out-of-sample OLS prediction	13	10.8	15	12.9	15	13.1	15	6.0
Simple mean combined forecast	14	12.4	6	6.5	8	7.4	8	7.9
SMA	7	7.3	8	7.7	12	10.1	13	10.5

Note: Ranking according to avg rank (lowest avg rank – 1, highest avg rank – 15). Average rank is calculated as a simple average of ranks (according to RMSE) across 20 stocks. Standard errors of averages are in Table A.12.

Chapter 6

Summary of aggregated results

Since Subsection 5.1.2 has a lot of results, it may be hard for a reader to see a full picture. Therefore, this chapter summarizes and compares all our results obtained in Subsection 5.1.2.

Table 6.3 lists models with most accurate results according to average RMSE, MAE and MAPE for all four volatility proxies, both training sub-samples and both forecast horizons. In Table 6.3, models written in regular font are ones from set of traditional time-series models (i.e. described in Section 4.1), models written in italics are ones from set of unconventional time-series models (i.e. described in Section 4.2), models written in bold are ones from set of forecast combination models (i.e. described in Section 4.3).

As you may see from Table 6.3, among models trained on medium training sub-sample, KNN regression on combined fit has most accurate results in many cases (10 out of 24) and if we add KNN regressions not on combined fit (i.e. KNN regressions on original data) results are most accurate in 13 cases out of 24. Additionally, if we focus on the most 'efficient' volatility proxy, which is Garman-Klass, we see that KNN regression on combined fit provides most accurate forecasts in 5 out of 6 cases. Only according to point estimate of average MAPE at horizon set to 10 periods ahead, SMA has more accurate results, however, KNN is ranked second and if we take into account standard errors (Table A.5), we can calculate that confidence intervals of average MAPE for both methods overlap significantly.

When we shrink our training sub-sample three times (from 750+ to 245+ observations), we see that KNN regression on combined fit is no longer a leading method (with being most accurate only in 1 case out of 24). We see that MLP models (either with one or two hidden layers) are most accurate in 10 cases out of 24. Additionally, we see that the simplest 'model', which is simple historical mean, is most accurate in 7 cases. Furthermore, for two most efficient volatility proxies (Garman-Klass and Parkinson), simple historical mean (which is a constant) is most accurate for both forecast horizons according to average RMSE and in addition, for horizon set to 5, it is most accurate according to average MAE. In other cases, for Parkinson and Garman-Klass, MLP models (either with 1 or 2 hidden layers) are most accurate.

In order to capture differences of forecasts performance of models trained on a short training sub-sample and a medium one, Table 6.1 is presented below. Table 6.1 displays percentage differences between most accurate forecasts of models trained on medium and short training sub-samples. We underlined values where most accurate methods were same for both training sub-samples. That is, for others, the most accurate forecast method changed.

Table 6.1: Relative difference of aggregated results in %.

	horizon = 5			horizon = 10		
Volatility proxy	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
Squared return	61.47	55.45	27.88	17.74	26.62	6.02
Absolute return	22.21	18.80	1.13	10.01	13.16	18.85
Parkinson	21.75	21.03	-4.50	10.67	9.10	-3.13
Garman Klass	21.80	23.07	-2.43	4.46	5.94	-14.22

Values for tables were calculated according to the following formula:

$$\frac{v_{short} - v_{medium}}{v_{short}} \cdot 100$$

where v_{short} is a value of a parameter according to a short training sub-sample and v_{medium} is a value of a parameter according to a medium training sub-sample. Thus, positive values mean that there was a forecast improvement with training sub-sample size increase since an error from a short sub-sample is larger than an error from a medium sub-sample.

In order to check whether relative differences presented in Table 6.1 are statistically significant, we present Table 6.2 where t statistics for differences are calculated according to the following formula:

$$t = \frac{\hat{\beta}_s - \hat{\beta}_m}{\sqrt{\hat{\sigma}_s^2 + \hat{\sigma}_m^2}}$$

where $\hat{\beta}_s$ is a parameter estimate from short sub-sample and $\hat{\beta}_m$ is a parameter estimate from a medium sub-sample and denominator is a sum of their standard deviations.

Table 6.2: T statistics of relative differences from Table 6.1.

	horizon = 5			horizon = 10		
Volatility proxy	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
Squared return	0.4642	0.4070	0.1117	0.1155	0.1917	0.0188
Absolute return	0.2865	0.2699	0.0086	0.1244	0.1912	0.1235
Parkinson	0.2370	0.2333	-0.0895	0.1260	0.1065	-0.0559
Garman Klass	0.2942	0.3213	-0.0469	0.0589	0.0773	-0.2720

From tables presented in this chapter, we see that if we have different length of training samples, we are likely to have different most accurate forecasting models. Only in 2 cases out of 24 the most accurate forecast was provided by same method. These are: (1) KNN regression on combined fit according to average MAPE for absolute return volatility proxy and horizon set to 5; and (2) MLP with 2 hidden layers according to average MAE for Parkinson volatility estimate. For other 22 cases, most accurate forecasts were provided by different models.

Likewise, from Table 6.3, it is seen that when trained on medium training sub-sample, for the most efficient volatility proxy, combination techniques (namely KNN regression on combined fit) provide most accurate results in 5 out of 6 cases. However, when trained on short training sub-sample, simple historical mean and MLP (either with 1 or 2 hidden layers) provide most accurate forecasts. It can be caused by the fact that simple historical mean on shorter sub-sample better reflects close future, however, we also see, that MLP models also perform well.

Finally, we see that though it may look like training sub-sample increase improves forecast accuracy (at least according to average MAE and RMSE), we need to check whether improvement is statistically significant and Table 6.2 suggests us that this is not the case[1].

With all the above we may draw some conclusions about our hypotheses:

> Hypothesis #1: Forecast from a combined model is more accurate than forecasts from conventional models or their extensions.

[1] For statistically significant difference in estimates, t statistic in Table 6.2 should be greater than 2.093 (in absolute value), which is a value from the t table for $\alpha = 0.05$ and 19 degrees of freedom.

> Hypothesis #2: Different training sub-samples require different models for same assets.

> Hypothesis #3: Training sub-sample increase deteriorates forecast accuracy.

Table 6.3 shows that hypothesis #1 is corroborated for Garman-Klass volatility proxy and medium training sub-sample where we see a clear dominance of combination technique (namely, KNN regression on combined fit). However, for other instances, it is harder to say whether combination techniques are superior to non-combination ones. It is all dependent on forecast horizon and forecast accuracy measure used. Additionally, for a short training sub-sample, we see that combination techniques are not superior to non-combination ones.

Likewise, Table 6.3 clearly supports hypothesis #2. We see that only in 2 cases (out of 24) same method remained to be the most accurate.

Finally, Table 6.2 clearly rejects hypothesis #3, since we do not see statistically significant forecast improvement in any of cases. It should be noted, that we compare the most accurate forecasts from two sub-samples that in 22 cases (out of 24) are provided by different models.

Table 6.3: Summary of aggregated results.

	Short training sub-sample					
	horizon = 5			horizon = 10		
Volatility proxy	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
Squared return	*MLP w/1 hidden layer*	EWMA	ARFIMA	*MLP w/1 hidden layer*	EWMA	EWMA
Absolute return	*MLP w/2 hidden layers*	*MLP w/2 hidden layers*	***KNN regression on combined fit***	Mean	NNAR	EWMA
Parkinson	Mean	Mean	*MLP w/1 hidden layer*	Mean	*MLP w/2 hidden layers*	*MLP w/1 hidden layer*
Garman Klass	Mean	Mean	*MLP /1 hidden layer*	Mean	*MLP w/ 2 hidden layers*	*MLP w/1 hidden layer*

	Medium training sub-sample					
	horizon = 5			horizon = 10		
Volatility proxy	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
Squared return	*KNN regression*	*KNN regression*	*KNN regression*	SMA	EWMA	EWMA
Absolute return	*NNAR*	***KNN regression on combined fit***	***KNN regression on combined fit***	ARIMA	***KNN regression on combined fit***	SMA
Parkinson	***KNN regression on combined fit***	***KNN regression on combined fit***	SMA	*MLP w/2 hidden layers*	*MLP w/2 hidden layers*	SMA
Garman Klass	***KNN regression on combined fit***	***KNN regression on combined fit***	***KNN regression on combined fit***	***KNN regression on combined fit***	***KNN regression on combined fit***	SMA

Chapter 7

Conclusion

The goal of the thesis was to challenge several issues a forecaster can encounter during selection of an optimal forecast. Namely, volatility proxy choice, training sample size choice and choice of a forecasting model. For these purposes, we carried out multiple forecasts for 4 volatility proxies (squared and absolute returns, Parkinson and Garman-Klass), 2 training samples (medium and short) and 15 forecasting models (10 standalone methods and 5 forecast combination methods). Forecasts were carried out for 20 stocks from different sectors and of different market capitalization size that were randomly selected. Forecasts quality was assessed by 3 metrics: RMSE, MAE and MAPE, and were assessed for 2 forecast horizons: one and two business weeks or 5 and 10 periods ahead. Forecasts were out-of-sample. In total, we generated 2400 forecasts[1] that needed to be aggregated, because otherwise it would be impossible to draw any general conclusions.

Aggregation was done on stocks level: we average results across 20 stocks and obtain 120 sets of forecast results. Afterwards, we focus only on most accurate forecast models but since we have 3 forecast accuracy assessment methods and 2 forecast horizons, we obtain one for each. So from 120 sets of forecasting results we go to 48 sets of results[2] that are presented in Chapter 6. And based on these results we are able to test our hypotheses.

We have 3 hypotheses and we draw general conclusions for two of them (hypotheses #2 and #3). Hypothesis #2 – different training sub-samples require different models for same assets – is not rejected, since from Table 6.3, we see

[1] 2400 = 4 x 2 x 15 x 20, i.e. number of volatility proxies x number of training samples x number of forecasting models x number of stocks.

[2] 48 = 4 x 2 x 3 x 2, i.e. number of volatility proxies x number of training samples x number of forecast accuracy metrics x number of forecast horizons.

that 'best' models change based on training sub-sample in 22 cases (out of 24). Hypothesis #3 – training sub-sample increase deteriorates forecast accuracy – is rejected, since from Table 6.2, we see that relative differences of 'best' models are statistically insignificant.

However, for hypothesis #1 – forecast from a combined model is more accurate than forecasts from conventional models or their extensions – that is actually our main hypothesis, we cannot clearly draw a conclusion because some narrowing is needed for this. For example, for Garman-Klass volatility proxy and medium training sample, hypothesis #1 holds, for other cases even more narrowing is needed. On short training sample, we see that forecast combination produces most accurate forecast only in one case (out of 24).

Likewise, we see that machine learning techniques (from Section 4.2) and simple forecasting methods (from Subsection 4.1.1) dominate traditional forecasting methods (from Subsection 4.1.2 and Subsection 4.1.3). Traditional forecasting methods are most accurate only in 2 cases (out of 48): (1) ARFIMA according to average MAPE for squared return, horizon set to 5 and trained on short sample and (2) ARIMA according to RMSE for absolute return, horizon set to 10 and trained on medium sample.

Additionally, we compared efficiency of volatility proxies. We define volatility proxy efficiency as level of its predictability according to average MAPE. We clearly see that squared and absolute return volatility proxies are much less efficient than Parkinson and Garman-Klass in all 24 cases. Whereas, distinction between efficiency of Parkinson and Garman-Klass is hard, though it is seen that Garman-Klass is marginally more efficient than Parkinson volatility proxy. Nonetheless, if we take into account standard errors, we may calculate that confidence intervals of average MAPE will overlap significantly.

We see there is plenty of room for future research in this field. OLS forecast combination, which is a base for our 'best' combination method – KNN regression on combined fit, could be improved if 'insignificant'[3] models used for OLS fit were dropped and OLS fit was retrained only with 'significant' models. Likewise, longer training samples could be utilized and results checked with short and medium training samples.

To conclude, we hope this thesis serves as a comprehensive guide in forecast combination and volatility estimation fields. We find our results interesting and revealing that machine learning and forecasting combination methods may yield superior results.

[3] 'Insignificant' models are ones that had statistically insignificant estimate in OLS fit.

Bibliography

Adya, M. & F. Collopy (1998): "How effective are neural networks at forecasting and prediction? a review and evaluation." *Journal of Forecasting* **17**: pp. 481–495.

Aiolfi, M., C. Capistran, & A. Timmermann (2010): "Forecast Combinations." *Working Papers 2010-04*, Banco de Mexico.

Aksu, C. & S. I. Gunter (1992): "An empirical analysis of the accuracy of sa, ols, erls and nrls combination forecasts." *International Journal of Forecasting* **8(1)**: pp. 27–43.

Alford, A. W. & J. R. Boatsman (1995): "Predicting long-term stock return volatility: Implications for accounting and valuation of equity derivatives." *The Accounting Review* **70(4)**: pp. 599–618.

Andersen, T., T. Bollerslev, & S. Lange (1999): "Forecasting financial market volatility: Sample frequency vis-a-vis forecast horizon." *Journal of Empirical Finance* **6(5)**: pp. 457–477.

Barrow, D. & N. Kourentzes (2018): "The impact of special days in call arrivals forecasting: A neural network approach to modelling special days." *European Journal of Operational Research* **264(3)**: pp. 967–977.

Bates, J. M. & C. W. J. Granger (1969): "The combination of forecasts." *OR* **20(4)**: pp. 451–468.

Ben Taieb, S., G. Bontempi, A. Atiya, & A. Sorjamaa (2012): "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition." *Expert Systems with Applications* **39(8)**: pp. 7067–7083.

Bollen, B. (2015): "What should the value of lambda be in the exponentially weighted moving average volatility model?" *Applied Economics* **47(8)**: pp. 853–860.

Bollerslev, T. (1986): "Generalized autoregressive conditional heteroskedasticity." *Journal of Econometrics* **31(3)**: pp. 307–327.

Boudoukh, J., M. Richardson, & R. F. Whitelaw (1997): "Investigation of a class of volatility estimators." *The Journal of Derivatives* **4(3)**: pp. 63–71.

Box, G. & D. Cox (1964): "An analysis of transformations." *Journal of the Royal Statistical Society, Series B* **26**: pp. 211–252.

Box, G. E. P. & G. M. Jenkins (1970): *Time Series Analysis: Forecasting and Control.* Holden-Day.

Connor, J., R. Martin, & L. Atlas (1994): "Recurrent neural networks and robust time series prediction." *IEEE Transactions on Neural Networks* **5(2)**: pp. 240–254.

Crone, S. F. & N. Kourentzes (2010): "Feature selection for time series prediction – a combined filter and wrapper approach for neural networks." *Neurocomputing* **73(10-12)**: pp. 1923–1936.

Cumby, R., S. Figlewski, & J. Hasbrouck (1993): "Forecasting volatilities and correlations with EGARCH models." *The Journal of Derivatives* **1(2)**: pp. 51–63.

Davidian, M. & R. J. Carroll (1987): "Variance function estimation." *Journal of the American Statistical Association* **82(400)**: pp. 1079–1091.

De Stefani, J., O. Caelen, D. Hattab, & G. Bontempi (2017): "Machine learning for multi-step ahead forecasting of volatility proxies." In "2nd Workshop on MIning DAta for financial applicationS (MIDAS)," .

Degiannakis, S. (2018): "Multiple days ahead realized volatility forecasting: Single, combined and average forecasts." *Global Finance Journal* **36**: pp. 41–61.

Diebold, F., A. Hickman, A. Inoue, & T. Schuermann (1998): "Scale models." *Risk Magazine* **11**: pp. 104–107.

DIEBOLD, F. X. (1988): "Serial correlation and the combination of forecasts." *Journal of Business & Economic Statistics* **6(1)**: pp. 105–111.

DING, Z., C. GRANGER, & R. ENGLE (1993): "A long memory property of stock market returns and a new model." *Journal of Empirical Finance* **1(1)**: pp. 83–106.

DONALDSON, R. G. & M. KAMSTRA (1997): "An artificial neural network-garch model for international stock return volatility." *Journal of Empirical Finance* **4(1)**: pp. 17–46.

ENGLE, R. F. & V. K. NG (1993): "Measuring and testing the impact of news on volatility." *The Journal of Finance* **48(5)**: pp. 1749–1778.

FIGLEWSKI, S. (1997): "Forecasting volatility." *Financial Markets, Institutions and Instruments* **6(1)**: pp. 1–88.

GAO, Y., C. ZHANG, & L. ZHANG (2012): "Comparison of garch models based on different distributions." *Journal of Computers* **7(8)**: pp. 1967–1973.

GARMAN, M. B. & M. J. KLASS (1980): "On the estimation of security price volatilities from historical data." *The Journal of Business* **53(1)**: pp. 67–78.

GHALANOS, A. (2017): *Introduction to the rugarch package.* Version 1.3-8.

GHALANOS, A. (2019): *rugarch: Univariate GARCH models.* R package version 1.4-1.

GLOSTEN, L. R., R. JAGANNATHAN, & D. E. RUNKLE (1993): "On the relation between the expected value and the volatility of the nominal excess return on stocks." *The Journal of Finance* **48(5)**: pp. 1779–1801.

GRANGER, C. W. J. & R. RAMANATHAN (1984): "Improved methods of combining forecasts." *Journal of Forecasting* **3(2)**: pp. 197–204.

HANS FRANSES, P. & D. VAN DIJK (1996): "Forecasting stock market volatility using (non-linear) garch models." *Journal of Forecasting* **15**: pp. 229–235.

HANSEN, P. R. & A. LUNDE (2005): "A forecast comparison of volatility models: does anything beat a GARCH(1,1)?" *Journal of Applied Econometrics* **20(7)**: pp. 873–889.

Hemanth Kumar, P. & S. Basavaraj Patil (2015): "Volatility forecasting using machine learning and time series techniques." *International Journal of Innovative Research in Computer and Communication Engineering* **3(9)**: pp. 8284–8292.

Hentschel, L. (1995): "All in the family nesting symmetric and asymmetric GARCH models." *Journal of Financial Economics* **39(1)**: pp. 71–104.

Higgins, M. L. & A. K. Bera (1992): "A class of nonlinear arch models." *International Economic Review* **33(1)**: pp. 137–158.

Hyndman, R., G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, & F. Yasmeen (2019): *forecast: Forecasting functions for time series and linear models.* R package version 8.7.

Hyndman, R. J. & G. Athanasopoulos (2018): *Forecasting: principles and practice.* OTexts.

Hyndman, R. J. & Y. Khandakar (2008): "Automatic time series forecasting: The forecast Package for R." *Journal of Statistical Software* **27(3)**.

Kourentzes, N. (2019): *nnfor: Time Series Forecasting with Neural Networks.* R package version 0.9.6.

Kourentzes, N., D. K. Barrow, & S. F. Crone (2014): "Neural network ensemble operators for time series forecasting." *Expert Systems with Applications* **41(9)**: pp. 4235–4244.

Kourentzes, N. & S. F. Crone (2010): "Frequency independent automatic input variable selection for neural networks for forecasting." In "The 2010 International Joint Conference on Neural Networks (IJCNN)," IEEE.

Lopez, J. A. (2001): "Evaluating the Predictive Accuracy of Volatility Models." *Journal of Forecasting* **20(2)**: pp. 87–109.

Makridakis, S. & R. L. Winkler (1983): "Averages of forecasts: Some empirical results." *Management Science* **29(9)**: pp. 987–996.

Martinez, F. (2019): *tsfknn: Time Series Forecasting Using Nearest Neighbors.* R package version 0.2.0.

Martinez, F., M. Pilar Frias, M. Dolores Perez, & A. Rivera Rivas (2017): "A methodology for applying k-nearest neighbor to time series forecasting." *Artificial Intelligence Review* .

McKenzie, M. D. (1999): "Power transformation and forecasting the magnitude of exchange rate changes." *International Journal of Forecasting* **15(1)**: pp. 49–55.

McMillan, D., A. Speight, & O. Apgwilym (2000): "Forecasting UK stock market volatility." *Applied Financial Economics* **10(4)**: pp. 435–448.

Newbold, P. & C. W. J. Granger (1974): "Experience with forecasting univariate time series and the combination of forecasts." *Journal of the Royal Statistical Society. Series A (General)* **137(2)**: pp. 131–165.

Parkinson, M. (1980): "The extreme value method for estimating the variance of the rate of return." *The Journal of Business* **53(1)**: pp. 61–65.

Poon, S.-H. & C. W. J. Granger (2003): "Forecasting volatility in financial markets: A review." *Journal of Economic Literature* **41(2)**: pp. 478–539.

RiskMetrics (1996): *RiskMetrics: Technical Document.* Morgan Guaranty Trust Company of New York, 4th edition.

Ryan, J. A. & J. M. Ulrich (2019): *quantmod: Quantitative Financial Modelling Framework.* R package version 0.4-14.

Schwert, G. W. (1990): "Stock volatility and the crash of '87." *The Review of Financial Studies* **3(1)**: pp. 77–102.

Taylor, S. J. (1986): *Modelling Financial Time Series.* Wiley.

Winkler, R. L. & S. Makridakis (1983): "The combination of forecasts." *Journal of the Royal Statistical Society. Series A (General)* **146(2)**: pp. 150–157.

Zakoian, J.-M. (1994): "Threshold heteroskedastic models." *Journal of Economic Dynamics and Control* **18(5)**: pp. 931–955.

Appendix A

Standard errors of averages

Standard errors are calculated for the following estimates: average RMSE, MAE and MAPE for each horizon, training sub-sample, volatility proxy and forecasting method based on sample of 20 observations (number of stocks). Formally, standard errors are defined as:

$$se = \frac{\sqrt{Var(X)}}{\sqrt{N}} \tag{A.1}$$

where X is a vector of point estimates and N is the length of vector X, in our case $N = 20$.

Table A.1: Standard errors of forecasts aggregation on medium training sub-sample and forecast horizon = 5 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy			**Absolute return volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	1.02E-03	1.02E-03	12659	1.22E-03	1.18E-03	99.3
ARIMA	1.51E-04	1.50E-04	7896	1.02E-03	9.07E-04	75.8
ARIMA on combined fit	1.58E-04	1.44E-04	16206	1.19E-03	1.11E-03	83.2
EWMA	5.90E-05	3.86E-05	7155	1.02E-03	8.56E-04	89.0
GARCH-Family	1.72E-03	1.72E-03	1209497	9.25E-04	7.19E-04	96.9
KNN regression	4.53E-05	3.70E-05	2760	1.21E-03	1.06E-03	117.4
KNN regression on combined fit	8.20E-05	6.41E-05	9289	7.95E-04	5.77E-04	47.1
Mean	4.31E-04	3.88E-04	15288	2.56E-03	2.61E-03	200.2
MLP w/ 1 hidden layer	1.73E-04	1.74E-04	17263	1.24E-03	1.20E-03	98.6
MLP w/ 2 hidden layers	3.12E-04	3.12E-04	129273	9.34E-04	8.49E-04	87.6
NNAR	1.60E-04	1.61E-04	16954	1.03E-03	9.97E-04	96.2
NNAR on combined fit	2.23E-04	1.82E-04	13790	1.05E-03	9.76E-04	59.6
Out-of-sample OLS prediction	4.08E-04	3.48E-04	13512	7.58E-04	6.64E-04	84.4
Simple mean combined forecast	2.78E-04	2.17E-04	11550	1.17E-03	1.03E-03	99.6
SMA	7.38E-05	6.55E-05	5005	1.07E-03	9.62E-04	55.7

Table A.2: Standard errors of forecasts aggregation on medium training sub-sample and forecast horizon = 5 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy			**Garman-Klass volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	6.82E-04	6.32E-04	3.589	5.57E-04	4.80E-04	3.030
ARIMA	5.95E-04	4.78E-04	2.102	5.62E-04	4.53E-04	2.377
ARIMA on combined fit	6.47E-04	5.58E-04	2.907	8.06E-04	7.15E-04	3.695
EWMA	7.53E-04	6.65E-04	3.333	7.97E-04	7.39E-04	3.458
GARCH-Family	8.94E-04	7.97E-04	4.194	9.11E-04	8.24E-04	4.292
KNN regression	6.10E-04	5.04E-04	2.544	5.87E-04	4.89E-04	4.150
KNN regression on combined fit	6.96E-04	6.15E-04	3.710	4.10E-04	3.58E-04	2.721
Mean	2.09E-03	2.10E-03	13.388	1.37E-03	1.32E-03	8.105
MLP w/ 1 hidden layer	8.84E-04	8.56E-04	5.250	8.12E-04	7.61E-04	4.769
MLP w/ 2 hidden layers	5.58E-04	4.68E-04	2.738	5.34E-04	4.67E-04	2.988
NNAR	5.52E-04	4.52E-04	2.313	5.97E-04	5.15E-04	3.370
NNAR on combined fit	6.81E-04	5.87E-04	3.713	2.04E-03	1.36E-03	7.967
Out-of-sample OLS prediction	5.26E-04	4.30E-04	3.309	5.74E-04	4.67E-04	3.247
Simple mean combined forecast	5.84E-04	4.75E-04	4.232	8.91E-04	7.85E-04	7.924
SMA	6.20E-04	5.15E-04	2.958	6.54E-04	5.79E-04	3.174

Table A.3: Standard errors of average ranks of forecasts on medium training sub-sample and forecast horizon = 5.

	Squared return volatility proxy	**Absolute return volatility proxy**	**Parkinson volatility proxy**	**Garman-Klass volatility proxy**
model	standard errors of avg rank			
ARFIMA	0.79	1.01	0.78	0.76
ARIMA	0.74	0.75	0.74	0.71
ARIMA on combined fit	0.81	0.73	0.64	0.75
EWMA	1.15	1.02	0.99	1.01
GARCH-Family	0.92	0.80	1.02	0.86
KNN regression	1.12	1.19	0.98	0.73
KNN regression on combined fit	0.70	1.02	0.93	1.04
Mean	1.03	1.09	1.19	1.25
MLP w/ 1 hidden layer	0.84	0.80	0.66	0.82
MLP w/ 2 hidden layers	0.80	1.13	1.00	1.10
NNAR	0.95	0.64	0.69	0.72
NNAR on combined fit	0.84	0.95	0.94	0.99
Out-of-sample OLS prediction	1.02	1.18	0.89	0.98
Simple mean combined forecast	0.70	0.33	0.50	0.59
SMA	1.06	1.07	1.08	0.97

Table A.4: Standard errors of forecasts aggregation on medium training sub-sample and forecast horizon = 10 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy			**Absolute return volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	1.02E-03	1.02E-03	14037	1.35E-03	1.02E-03	99.6
ARIMA	1.69E-04	1.54E-04	9077	1.19E-03	8.44E-04	78.6
ARIMA on combined fit	1.76E-04	1.48E-04	10286	1.22E-03	9.48E-04	86.1
EWMA	1.29E-04	5.70E-05	7439	1.42E-03	9.78E-04	98.9
GARCH-Family	1.73E-03	1.74E-03	606165	1.24E-03	7.50E-04	108.2
KNN regression	1.50E-04	7.35E-05	10795	1.51E-03	9.75E-04	95.9
KNN regression on combined fit	1.36E-04	7.87E-05	8245	1.14E-03	6.48E-04	64.9
Mean	4.57E-04	4.31E-04	9334	2.18E-03	2.15E-03	163.1
MLP w/ 1 hidden layer	1.85E-04	1.64E-04	15961	1.36E-03	1.03E-03	100.4
MLP w/ 2 hidden layers	3.07E-04	3.06E-04	64510	1.23E-03	7.75E-04	92.2
NNAR	1.78E-04	1.56E-04	14767	1.31E-03	9.22E-04	96.0
NNAR on combined fit	2.03E-04	1.59E-04	12275	3.40E-03	1.63E-03	118.5
Out-of-sample OLS prediction	4.01E-04	3.66E-04	11215	1.21E-03	7.83E-04	79.1
Simple mean combined forecast	2.56E-04	2.05E-04	9287	1.28E-03	8.94E-04	85.8
SMA	1.25E-04	7.84E-05	9191	1.25E-03	8.37E-04	73.0

Table A.5: Standard errors of forecasts aggregation on medium training sub-sample and forecast horizon = 10 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy			**Garman-Klass volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	7.63E-04	6.56E-04	3.027	7.05E-04	6.08E-04	2.623
ARIMA	5.90E-04	4.69E-04	2.474	5.75E-04	4.52E-04	2.362
ARIMA on combined fit	6.42E-04	5.35E-04	3.128	7.19E-04	5.75E-04	2.573
EWMA	9.59E-04	8.41E-04	4.581	9.41E-04	8.61E-04	4.630
GARCH-Family	8.33E-04	7.20E-04	4.236	8.57E-04	7.52E-04	4.346
KNN regression	6.77E-04	5.58E-04	1.631	6.01E-04	4.95E-04	2.198
KNN regression on combined fit	6.91E-04	5.54E-04	3.416	5.19E-04	4.29E-04	1.864
Mean	1.91E-03	1.89E-03	14.872	1.39E-03	1.36E-03	11.437
MLP w/ 1 hidden layer	9.15E-04	8.30E-04	4.238	9.04E-04	8.04E-04	3.980
MLP w/ 2 hidden layers	6.34E-04	5.07E-04	2.389	6.14E-04	5.23E-04	2.441
NNAR	7.32E-04	6.15E-04	3.090	7.42E-04	6.59E-04	3.557
NNAR on combined fit	6.78E-04	5.73E-04	2.443	1.48E-03	9.26E-04	4.222
Out-of-sample OLS prediction	6.96E-04	5.75E-04	2.868	6.85E-04	5.75E-04	3.052
Simple mean combined forecast	6.31E-04	5.23E-04	3.566	8.43E-04	7.19E-04	6.634
SMA	6.40E-04	4.97E-04	3.068	6.46E-04	5.43E-04	2.703

Table A.6: Standard errors of average ranks of forecasts on medium training sub-sample and forecast horizon = 10.

	Squared return volatility proxy	**Absolute return volatility proxy**	**Parkinson volatility proxy**	**Garman-Klass volatility proxy**
model	standard errors of avg rank			
ARFIMA	0.92	0.86	0.82	0.74
ARIMA	0.78	0.83	0.61	0.61
ARIMA on combined fit	0.94	0.64	0.67	0.68
EWMA	1.00	0.99	0.79	0.97
GARCH-Family	1.31	0.87	0.92	1.05
KNN regression	1.09	1.04	1.00	0.92
KNN regression on combined fit	0.76	0.96	1.05	1.00
Mean	0.83	0.97	1.11	1.19
MLP w/ 1 hidden layer	0.80	0.84	0.86	1.01
MLP w/ 2 hidden layers	0.71	1.08	1.11	0.88
NNAR	0.88	0.76	0.64	0.73
NNAR on combined fit	1.01	0.98	1.04	0.97
Out-of-sample OLS prediction	0.95	1.06	0.79	0.66
Simple mean combined forecast	0.94	0.52	0.53	0.57
SMA	1.01	1.03	0.93	0.92

Table A.7: Standard errors of forecasts aggregation on short training sub-sample and forecast horizon = 5 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy			**Absolute return volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	1.90E-04	1.13E-04	5595	1.89E-03	1.35E-03	94
ARIMA	1.66E-04	1.30E-04	9199	1.95E-03	1.46E-03	95
ARIMA on combined fit	1.89E-04	1.47E-04	10163	2.05E-03	1.53E-03	91
EWMA	1.94E-04	1.16E-04	5573	2.15E-03	1.46E-03	98
GARCH-Family	1.96E-03	1.91E-03	787970	2.24E-03	1.78E-03	118
KNN regression	1.93E-04	1.21E-04	8612	2.13E-03	1.56E-03	90
KNN regression on combined fit	1.91E-04	1.24E-04	7535	2.07E-03	1.57E-03	75
Mean	1.64E-04	1.21E-04	9182	1.87E-03	1.33E-03	93
MLP w/ 1 hidden layer	1.64E-04	1.19E-04	9071	2.02E-03	1.45E-03	97
MLP w/ 2 hidden layers	1.64E-04	1.19E-04	9196	1.78E-03	1.25E-03	94
NNAR	1.68E-04	1.20E-04	9126	1.86E-03	1.31E-03	96
NNAR on combined fit	2.72E-04	2.34E-04	12817	2.43E-03	1.93E-03	83
Out-of-sample OLS prediction	2.40E-04	2.24E-04	41100	3.18E-03	2.78E-03	198
Simple mean combined forecast	2.25E-04	2.13E-04	86113	2.17E-03	1.56E-03	103
SMA	1.82E-04	1.19E-04	12518	2.01E-03	1.42E-03	103

Table A.8: Standard errors of forecasts aggregation on short training sub-sample and forecast horizon = 5 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy			**Garman-Klass volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	1.07E-03	8.17E-04	1.75	8.97E-04	7.77E-04	1.92
ARIMA	1.15E-03	9.37E-04	3.68	8.69E-04	7.29E-04	3.19
ARIMA on combined fit	1.19E-03	9.56E-04	2.37	8.88E-04	7.67E-04	3.79
EWMA	1.37E-03	1.18E-03	2.98	1.07E-03	9.91E-04	3.02
GARCH-Family	1.32E-03	1.10E-03	5.06	9.94E-04	8.26E-04	4.53
KNN regression	1.19E-03	1.01E-03	3.20	9.95E-04	8.19E-04	2.89
KNN regression on combined fit	1.33E-03	1.10E-03	4.04	9.69E-04	8.71E-04	4.32
Mean	9.99E-04	7.47E-04	2.26	7.42E-04	5.80E-04	1.83
MLP w/ 1 hidden layer	1.13E-03	8.76E-04	1.90	8.11E-04	6.62E-04	1.69
MLP w/ 2 hidden layers	1.00E-03	7.39E-04	1.94	7.58E-04	5.83E-04	1.81
NNAR	1.16E-03	9.63E-04	3.27	8.84E-04	7.33E-04	2.30
NNAR on combined fit	1.18E-03	1.01E-03	4.57	9.12E-04	7.89E-04	4.04
Out-of-sample OLS prediction	1.69E-03	1.56E-03	7.67	2.61E-03	2.61E-03	16.20
Simple mean combined forecast	1.27E-03	1.03E-03	2.57	9.60E-04	8.05E-04	1.89
SMA	1.34E-03	1.18E-03	3.66	1.07E-03	9.90E-04	2.90

Table A.9: Standard errors of average ranks of forecasts on short training sub-sample and forecast horizon = 5.

	Squared return volatility proxy	**Absolute return volatility proxy**	**Parkinson volatility proxy**	**Garman-Klass volatility proxy**
model	standard errors of avg rank			
ARFIMA	0.68	0.75	0.59	0.70
ARIMA	0.56	0.60	0.77	0.65
ARIMA on combined fit	0.88	0.94	0.83	0.89
EWMA	1.04	0.77	0.92	0.57
GARCH-Family	0.47	1.03	1.03	1.29
KNN regression	0.85	0.94	0.97	1.00
KNN regression on combined fit	0.88	1.21	0.86	1.02
Mean	0.62	0.66	0.58	0.72
MLP w/ 1 hidden layer	0.72	0.50	0.77	0.53
MLP w/ 2 hidden layers	0.75	0.78	0.62	0.56
NNAR	0.83	0.73	0.79	0.69
NNAR on combined fit	1.05	1.21	0.98	1.10
Out-of-sample OLS prediction	0.99	0.77	0.83	0.68
Simple mean combined forecast	0.90	0.84	0.67	0.80
SMA	0.71	0.90	0.92	0.90

Table A.10: Standard errors of forecasts aggregation on short training sub-sample and forecast horizon = 10 for Squared return and Absolute return volatility proxies.

	Squared return volatility proxy			**Absolute return volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	1.53E-04	9.36E-05	4404	1.59E-03	1.19E-03	89
ARIMA	1.44E-04	1.20E-04	10060	1.70E-03	1.31E-03	90
ARIMA on combined fit	1.65E-04	1.31E-04	15010	1.76E-03	1.33E-03	75
EWMA	1.57E-04	9.22E-05	7270	1.81E-03	1.25E-03	102
GARCH-Family	2.64E-03	2.47E-03	467976	2.37E-03	2.02E-03	129
KNN regression	1.56E-04	9.49E-05	7603	1.81E-03	1.30E-03	75
KNN regression on combined fit	1.77E-04	1.17E-04	9224	1.87E-03	1.37E-03	79
Mean	1.38E-04	1.07E-04	10009	2.95E-03	2.66E-03	199
MLP w/ 1 hidden layer	1.37E-04	1.06E-04	9994	1.59E-03	1.19E-03	85
MLP w/ 2 hidden layers	1.37E-04	1.05E-04	11720	1.80E-03	1.36E-03	99
NNAR	1.41E-04	1.05E-04	9565	1.68E-03	1.30E-03	89
NNAR on combined fit	2.14E-04	1.66E-04	15642	1.76E-03	1.30E-03	83
Out-of-sample OLS prediction	2.41E-04	2.30E-04	23258	1.58E-03	1.18E-03	92
Simple mean combined forecast	2.93E-04	2.73E-04	50342	2.07E-03	1.60E-03	92
SMA	1.50E-04	9.64E-05	10781	1.78E-03	1.29E-03	85

Table A.11: Standard errors of forecasts aggregation on short training sub-sample and forecast horizon = 10 for Parkinson and Garman-Klass volatility proxies.

	Parkinson volatility proxy			**Garman-Klass volatility proxy**		
model	avg RMSE	avg MAE	avg MAPE	avg RMSE	avg MAE	avg MAPE
ARFIMA	8.7E-04	6.7E-04	2.67	8.3E-04	7.1E-04	2.00
ARIMA	9.9E-04	8.4E-04	4.54	8.6E-04	7.7E-04	3.93
ARIMA on combined fit	9.7E-04	7.8E-04	2.75	8.6E-04	7.9E-04	4.01
EWMA	1.2E-03	1.1E-03	3.37	1.1E-03	1.0E-03	3.18
GARCH-Family	1.8E-03	1.5E-03	12.57	1.7E-03	1.4E-03	11.08
KNN regression	9.6E-04	7.8E-04	3.65	8.3E-04	7.1E-04	2.93
KNN regression on combined fit	1.0E-03	8.3E-04	3.76	9.0E-04	7.9E-04	3.60
Mean	8.3E-04	6.3E-04	3.13	6.9E-04	5.7E-04	2.48
MLP w/ 1 hidden layer	9.1E-04	7.0E-04	2.73	7.2E-04	6.0E-04	2.05
MLP w/ 2 hidden layers	8.3E-04	6.3E-04	2.94	6.9E-04	5.6E-04	2.39
NNAR	9.2E-04	7.4E-04	3.21	8.0E-04	6.9E-04	2.20
NNAR on combined fit	1.1E-03	9.6E-04	5.83	8.2E-04	6.9E-04	3.22
Out-of-sample OLS prediction	1.6E-03	1.5E-03	8.57	2.6E-03	2.6E-03	17.28
Simple mean combined forecast	1.0E-03	8.5E-04	3.44	8.9E-04	7.7E-04	2.39
SMA	1.1E-03	1.0E-03	3.15	1.0E-03	9.6E-04	2.34

Table A.12: Standard errors of average ranks of forecasts on short training sub-sample and forecast horizon = 10.

model	**Squared return volatility proxy**	**Absolute return volatility proxy**	**Parkinson volatility proxy**	**Garman-Klass volatility proxy**
	standard errors of avg rank			
ARFIMA	0.70	0.73	0.59	0.70
ARIMA	0.55	0.52	0.84	0.78
ARIMA on combined fit	0.87	0.82	0.88	0.86
EWMA	0.88	0.71	0.89	0.86
GARCH-Family	0.50	1.19	1.40	1.44
KNN regression	0.70	0.89	0.81	0.87
KNN regression on combined fit	1.05	1.06	0.90	0.84
Mean	0.61	0.61	0.47	0.63
MLP w/ 1 hidden layer	0.68	0.68	0.69	0.51
MLP w/ 2 hidden layers	0.71	0.78	0.48	0.51
NNAR	0.64	0.74	0.74	0.67
NNAR on combined fit	0.93	1.08	0.68	0.99
Out-of-sample OLS prediction	0.88	0.93	0.75	0.62
Simple mean combined forecast	0.89	0.69	0.64	0.65
SMA	0.84	1.04	0.90	0.81